

Fig. 1
Prior Art

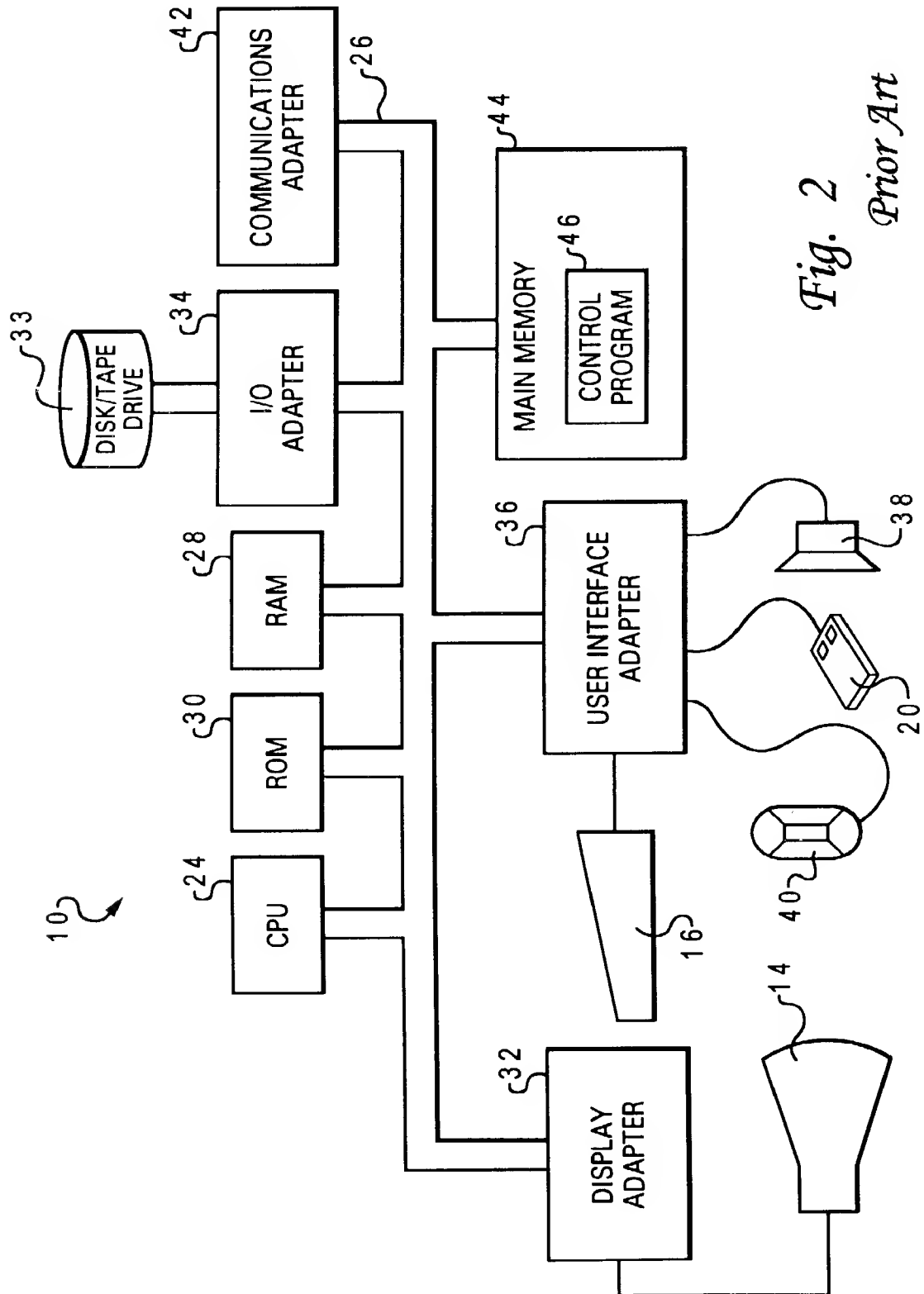


Fig. 2
Prior Art

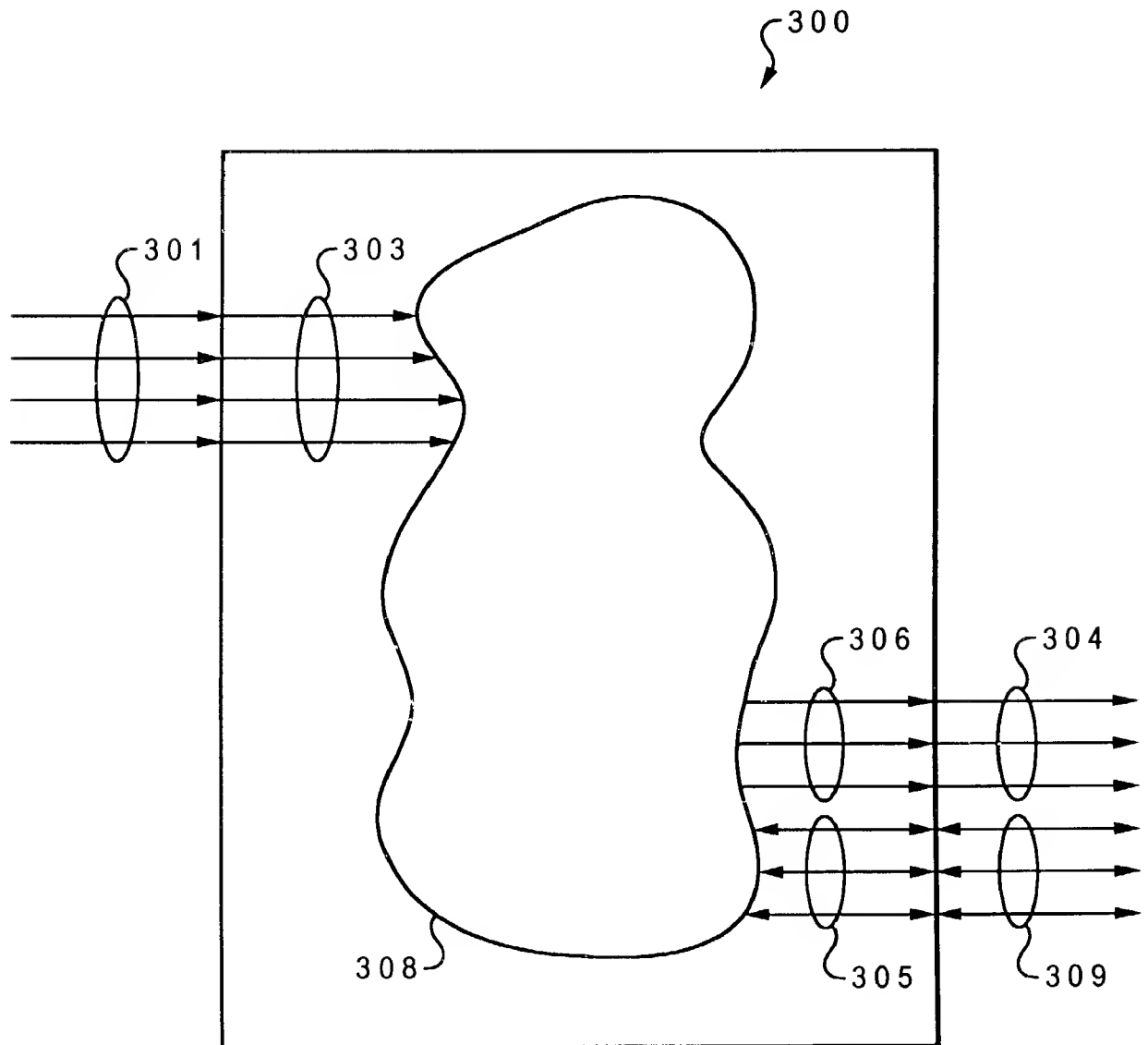
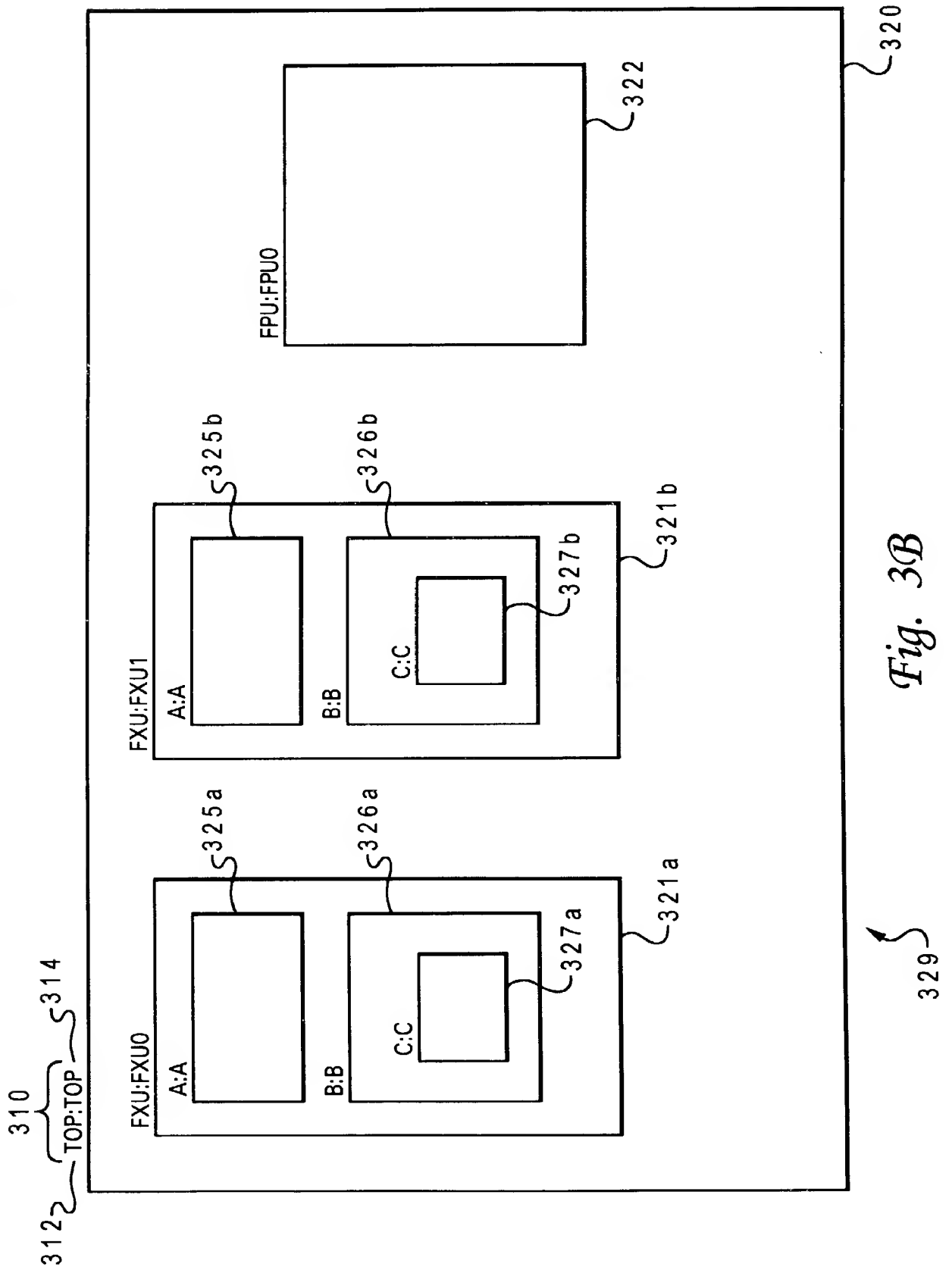


Fig. 3A



5/62

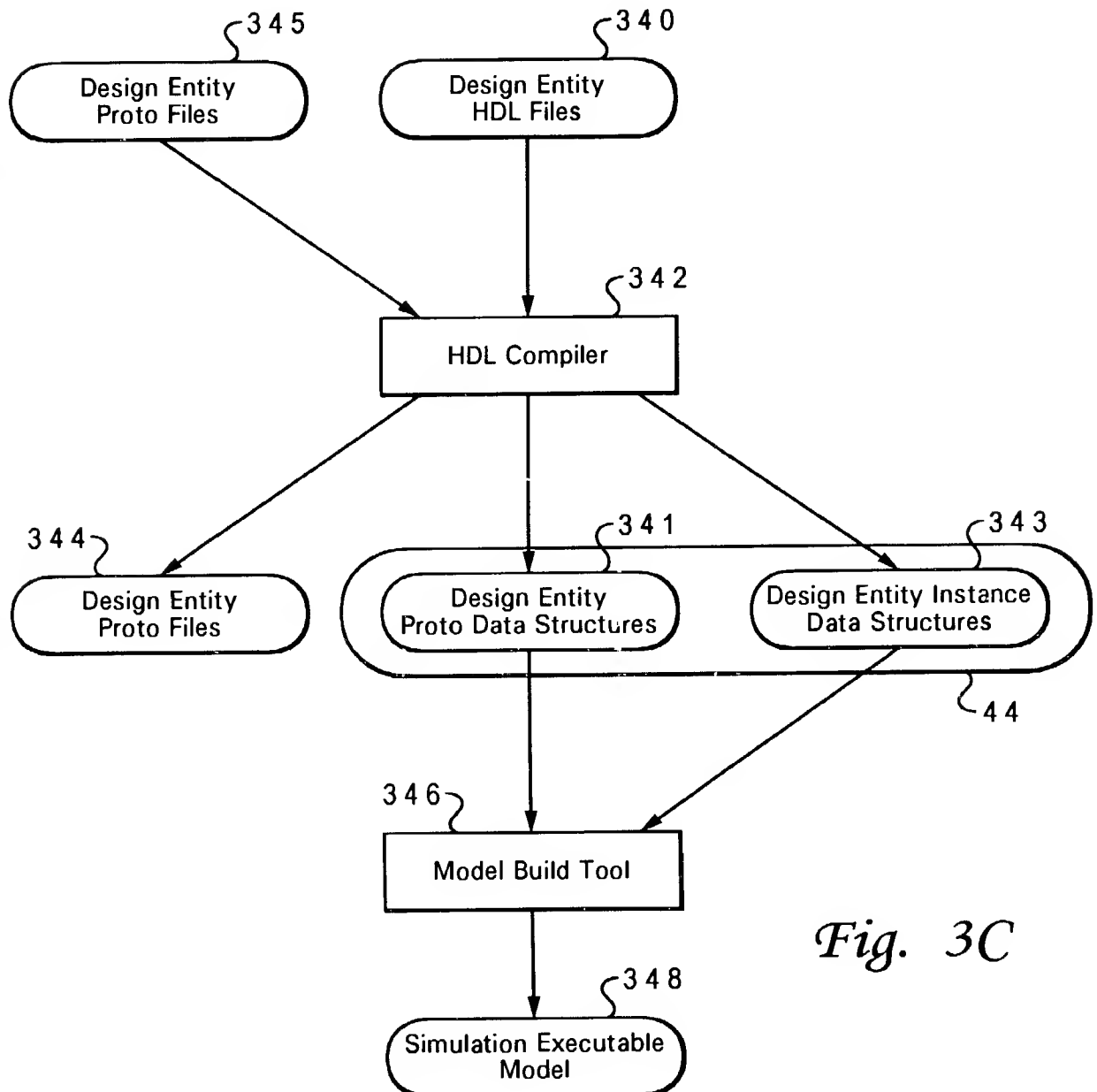


Fig. 3C

6/62

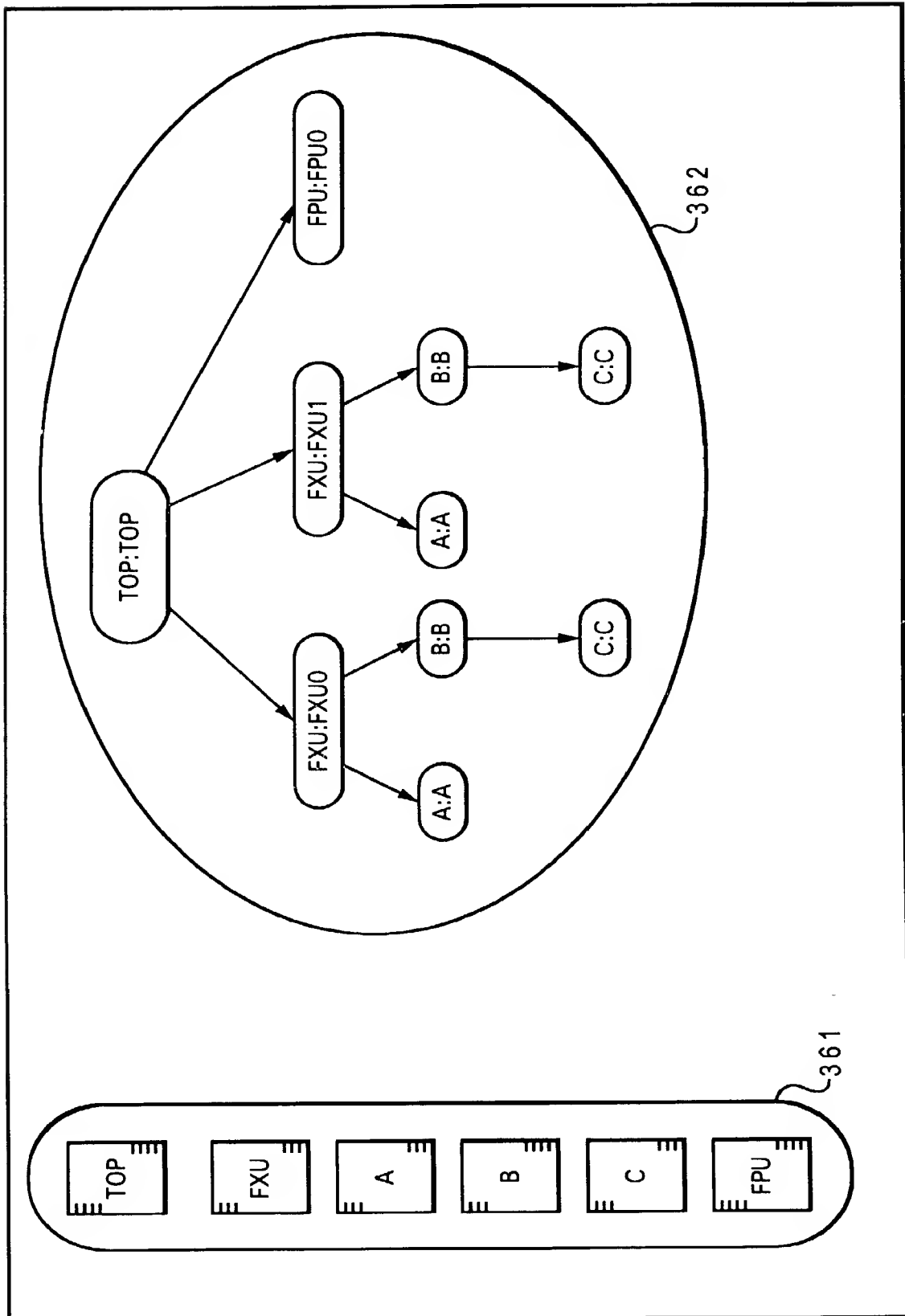


Fig. 3D

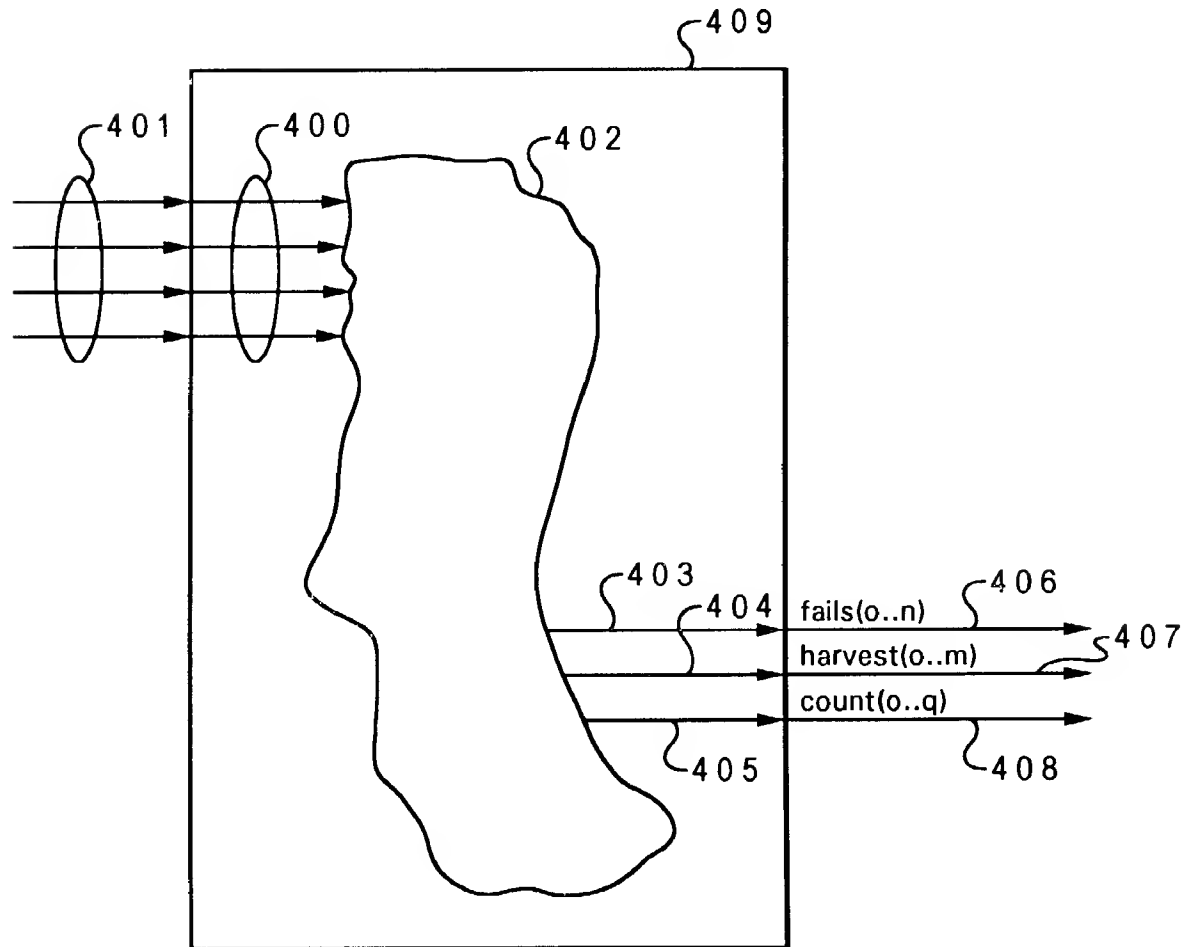


Fig. 4A

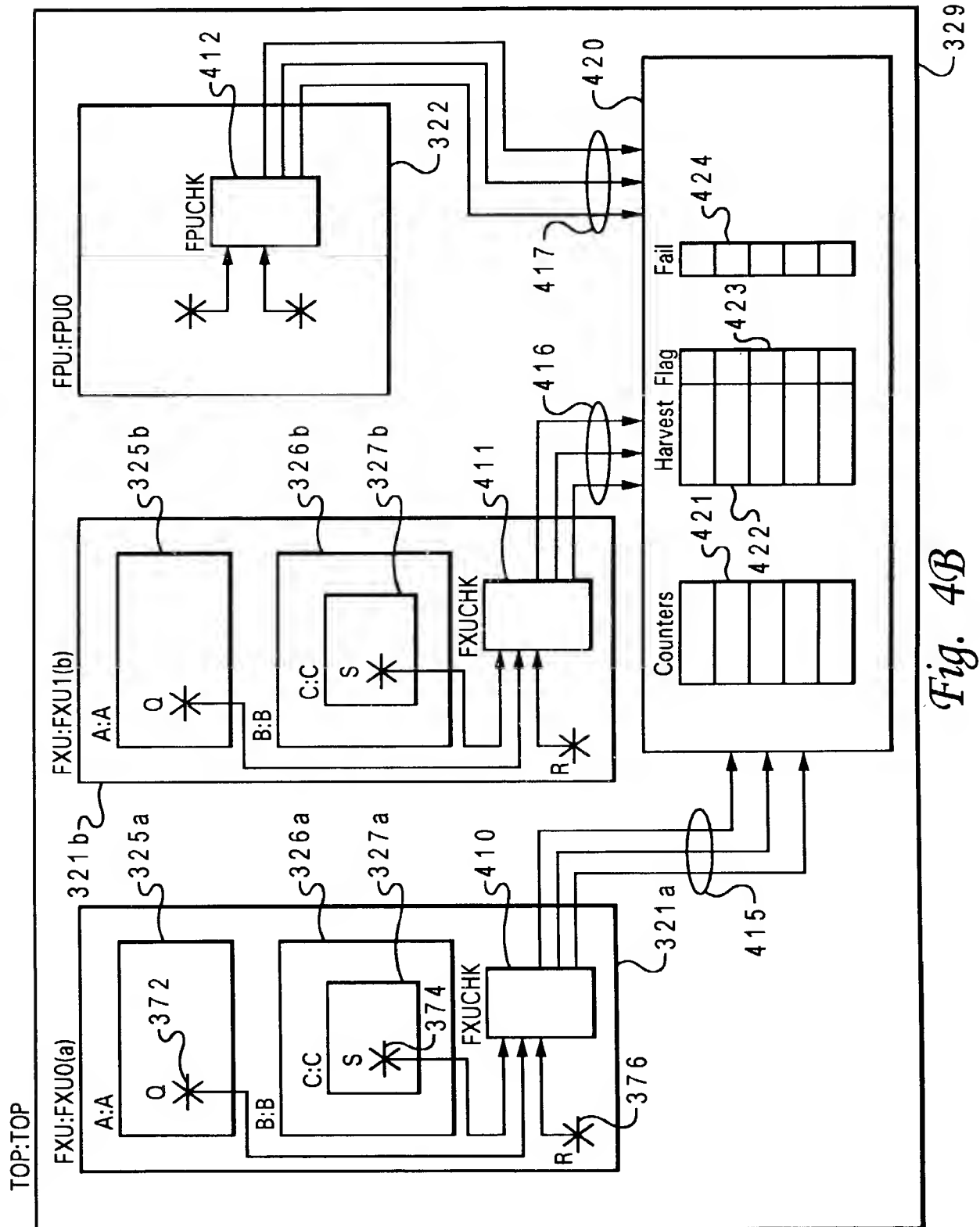


Fig. 4B

9/62

ENTITY FXUCHK IS

```

    PORT(  S_IN      :    IN std_ulogic;
           Q_IN      :    IN std_ulogic;
           R_IN      :    IN std_ulogic;
           clock      :    IN std_ulogic;
           fails      :    OUT std_ulogic_vector(0 to 1);
           counts     :    OUT std_ulogic_vector(0 to 2);
           harvests   :    OUT std_ulogic_vector(0 to 1);
    );

```

4 5 0

```

4 5 2 { --!! BEGIN
      --!! Design Entity: FXU;

```

```

4 5 3 { --!! Inputs
      --!! S_IN      =>    B.C.S;
      --!! Q_IN      =>    A.Q;
      --!! R_IN      =>    R;
      --!! CLOCK     =>    clock;
      --!! End Inputs

```

```

4 5 4 { --!! Fail Outputs;
      --!! 0 : "Fail message for failure event 0";
      --!! 1 : "Fail message for failure event 1";
      --!! End Fail Outputs;

```

```

4 5 5 { --!! Count Outputs;
      --!! 0 : <event0> clock;
      --!! 1 : <event1> clock;
      --!! 2 : <event2> clock;
      --!! End Count Outputs;

```

```

4 5 6 { --!! Harvest Outputs;
      --!! 0 : "Message for harvest event 0";
      --!! 1 : "Message for harvest event 1";
      --!! End Harvest Outputs;

```

```

4 5 7 { --!! End;

```

4 5 1

4 4 0

ARCHITECTURE example of FXUCHK IS

```

BEGIN

```

```

    ... HDL code for entity body section ...

```

```

END;

```

4 5 8

Fig. 4C

10/62

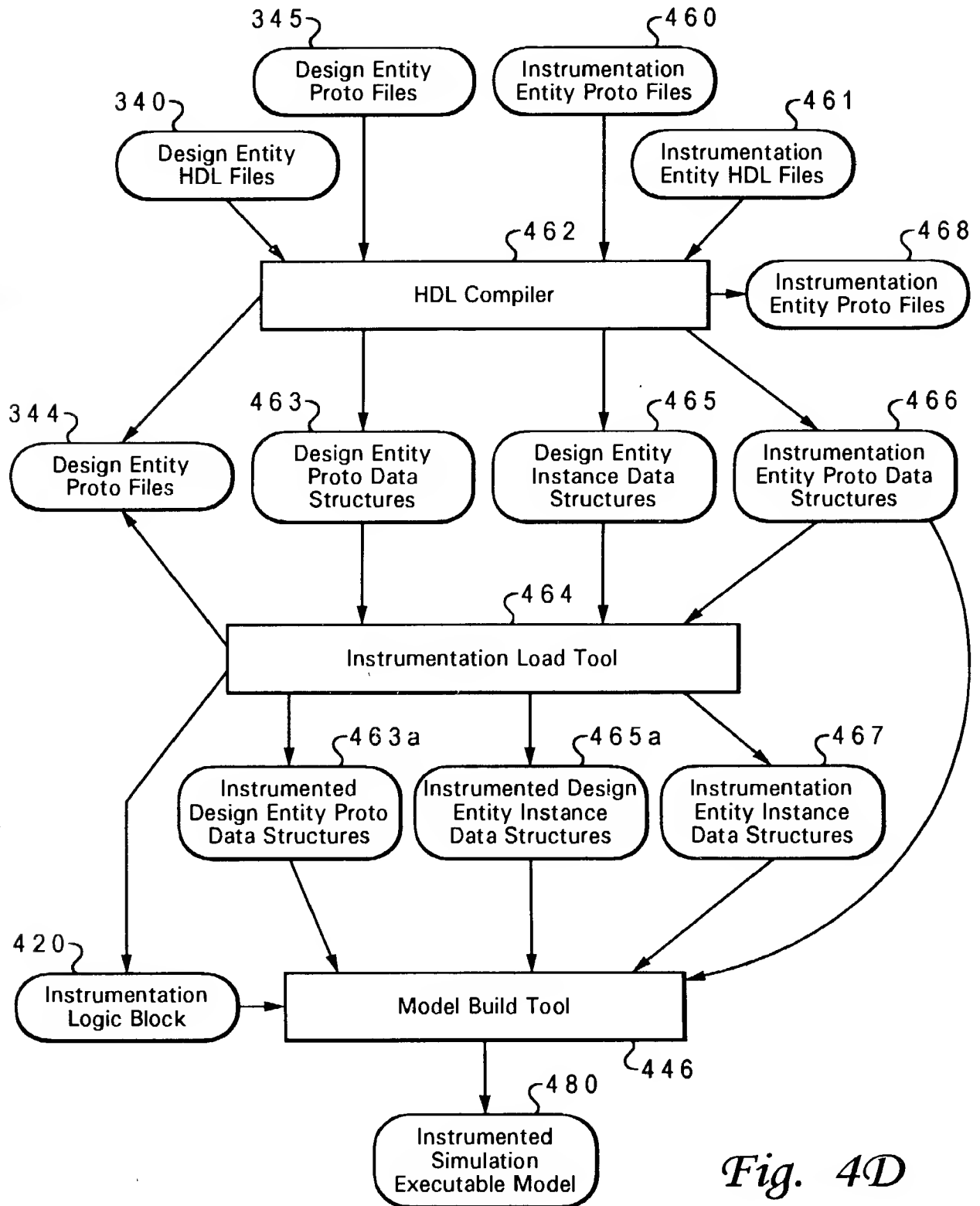


Fig. 4D

11/62

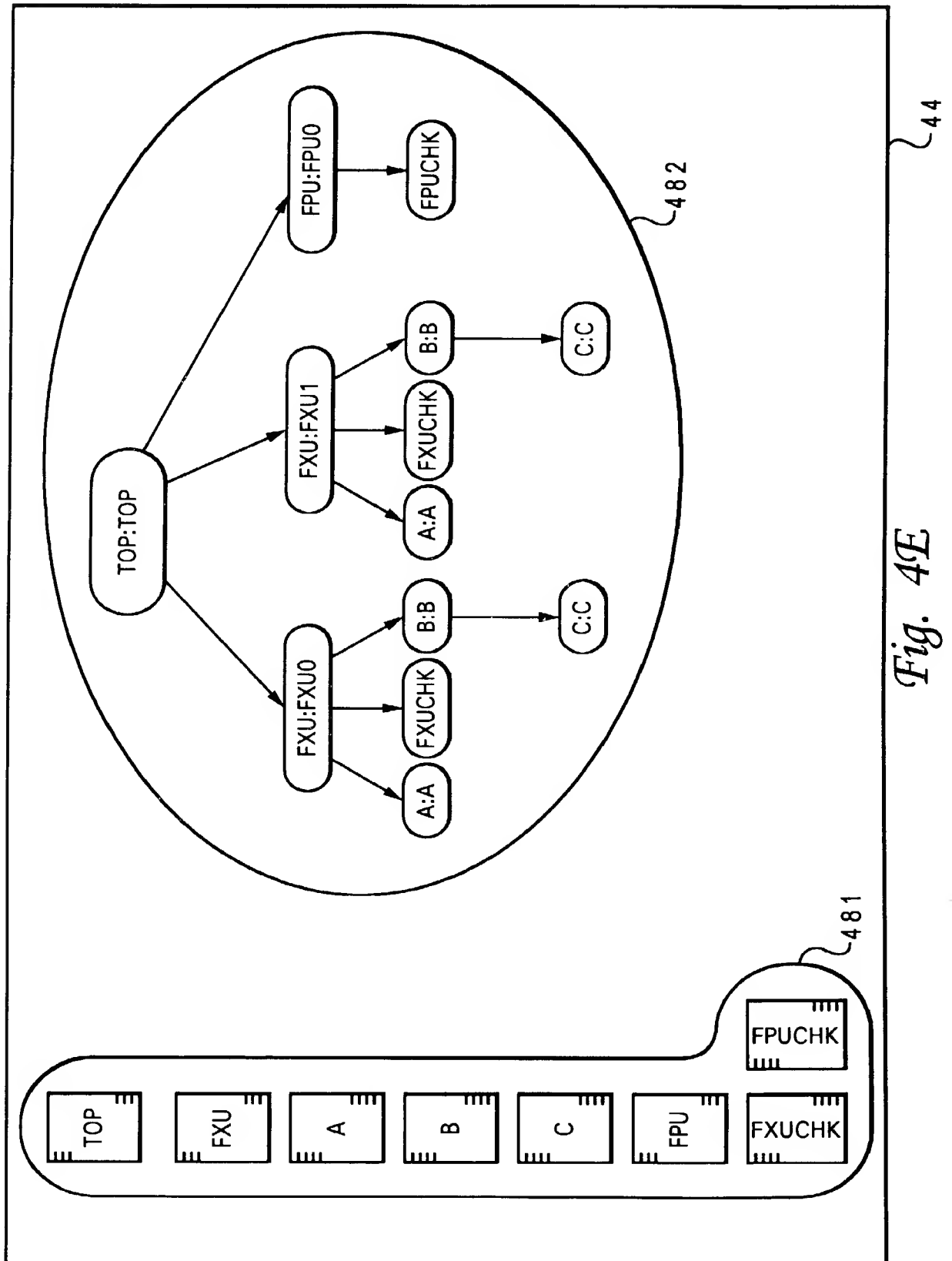
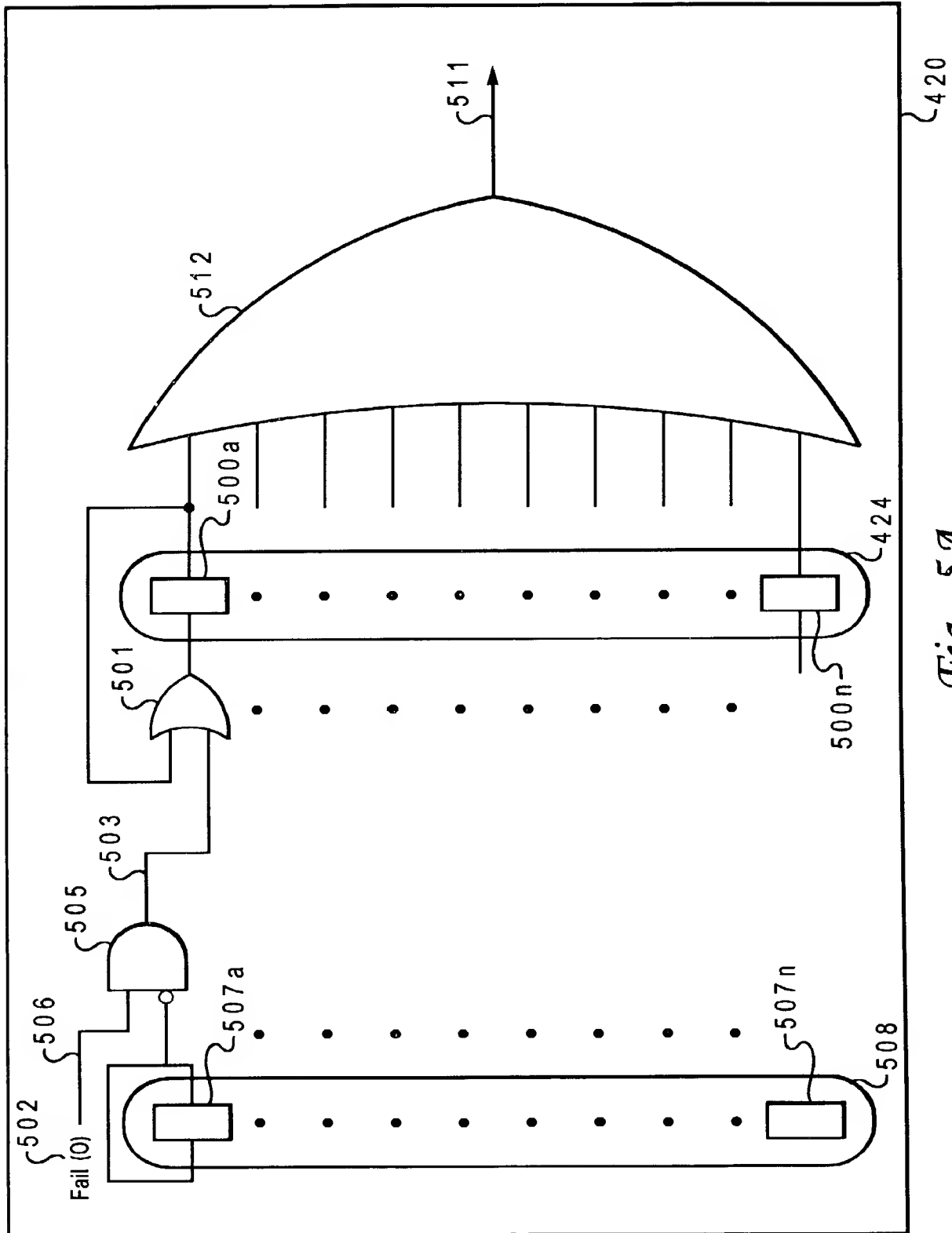


Fig. 4E

12/62



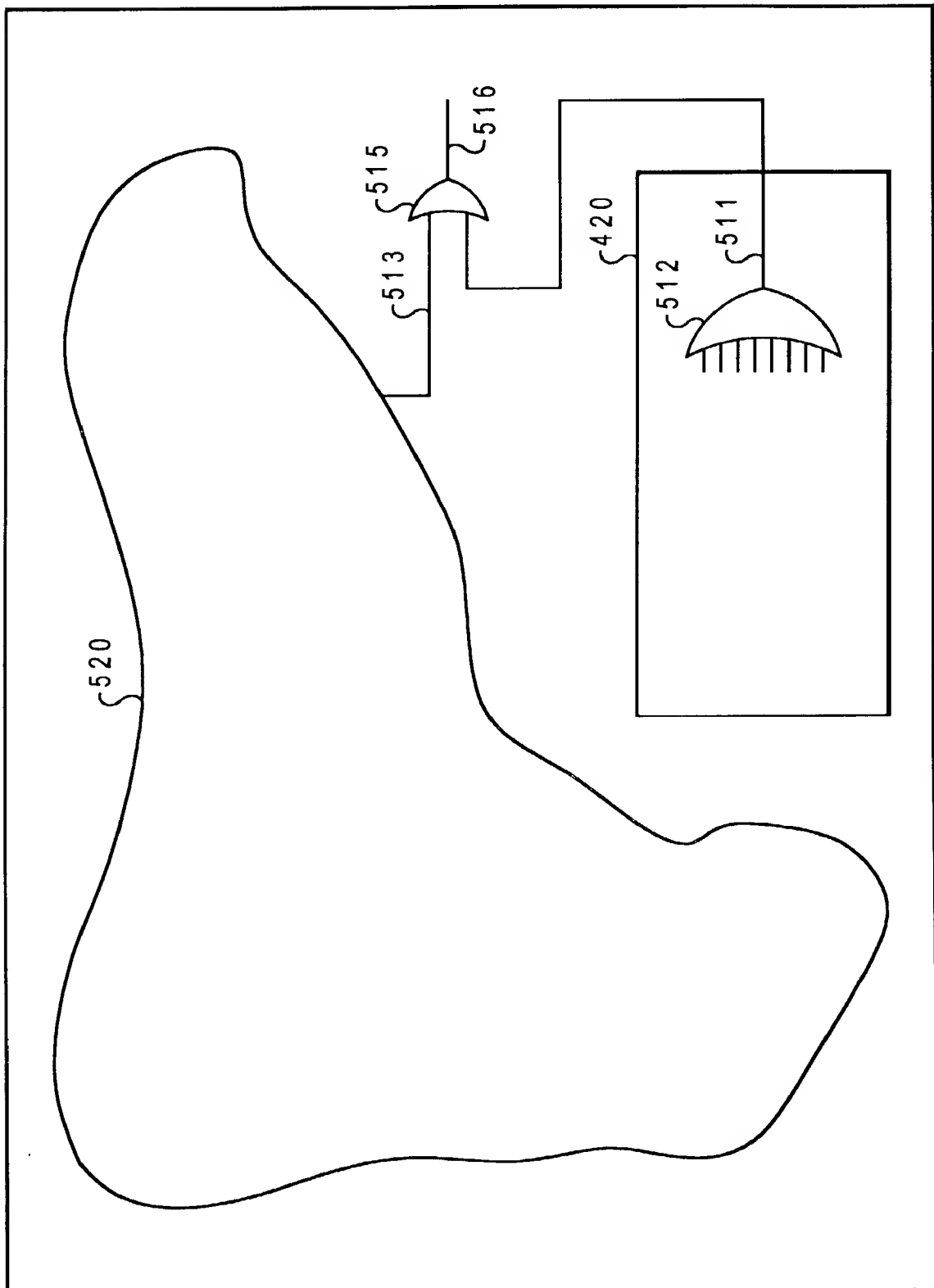


Fig. 5B

14/62

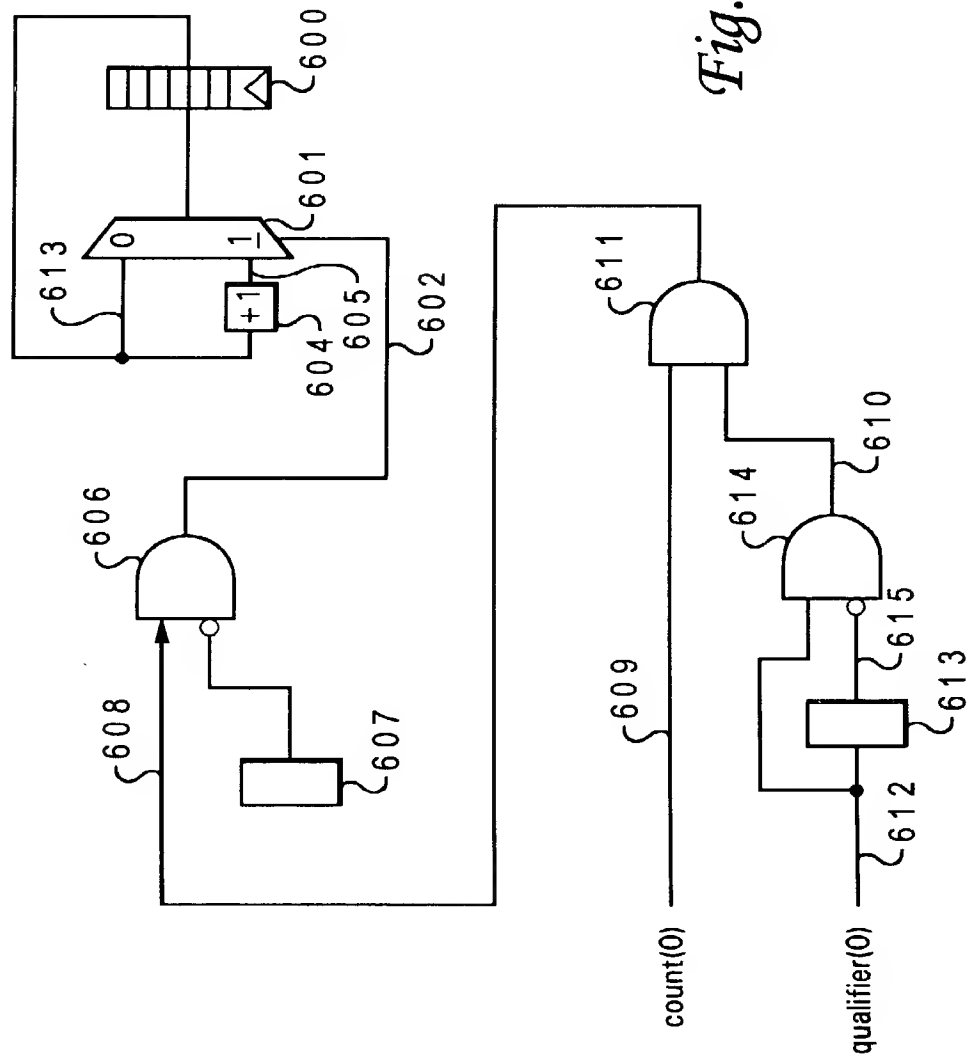


Fig. 6A

15/62

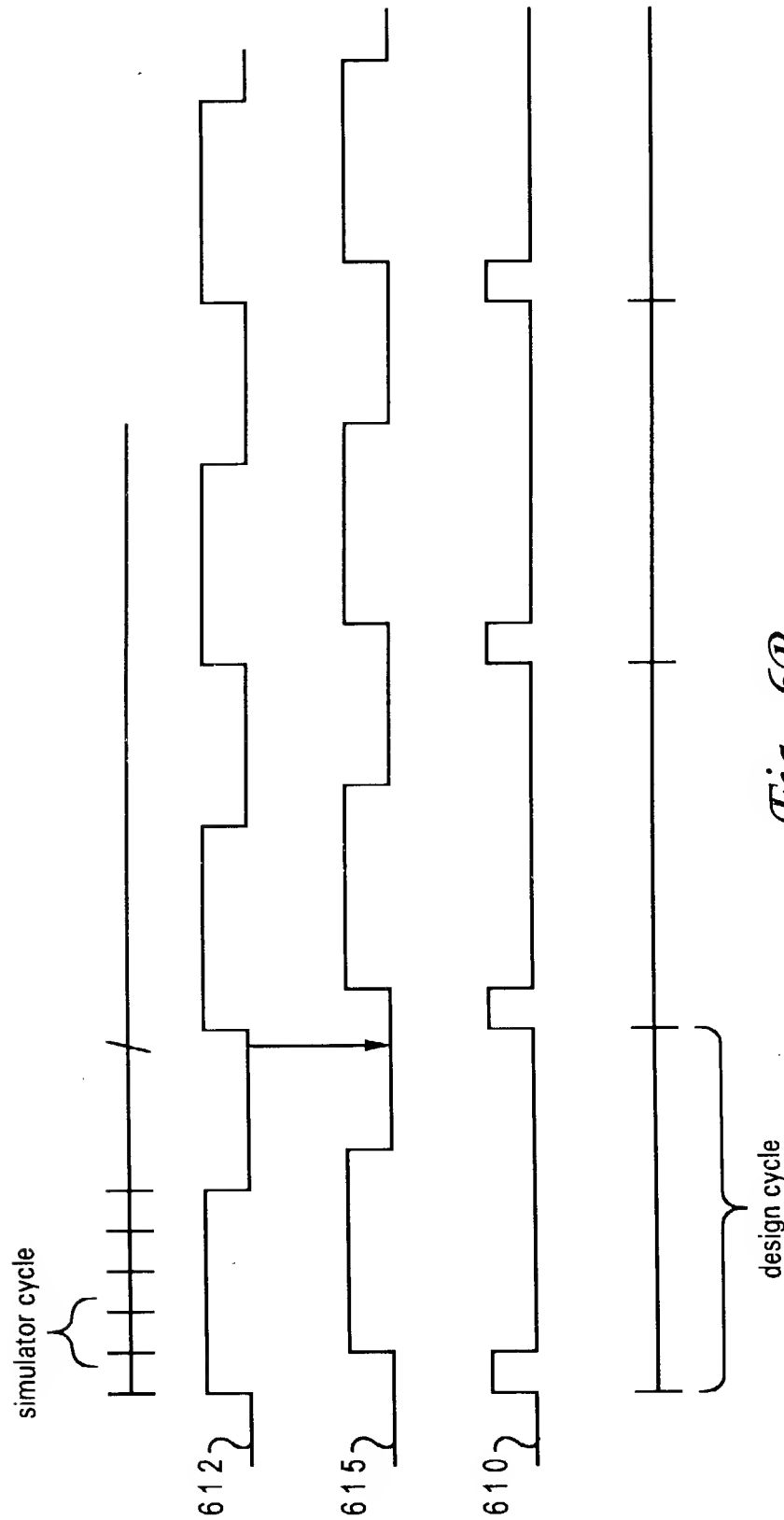


Fig. 6B

16/62

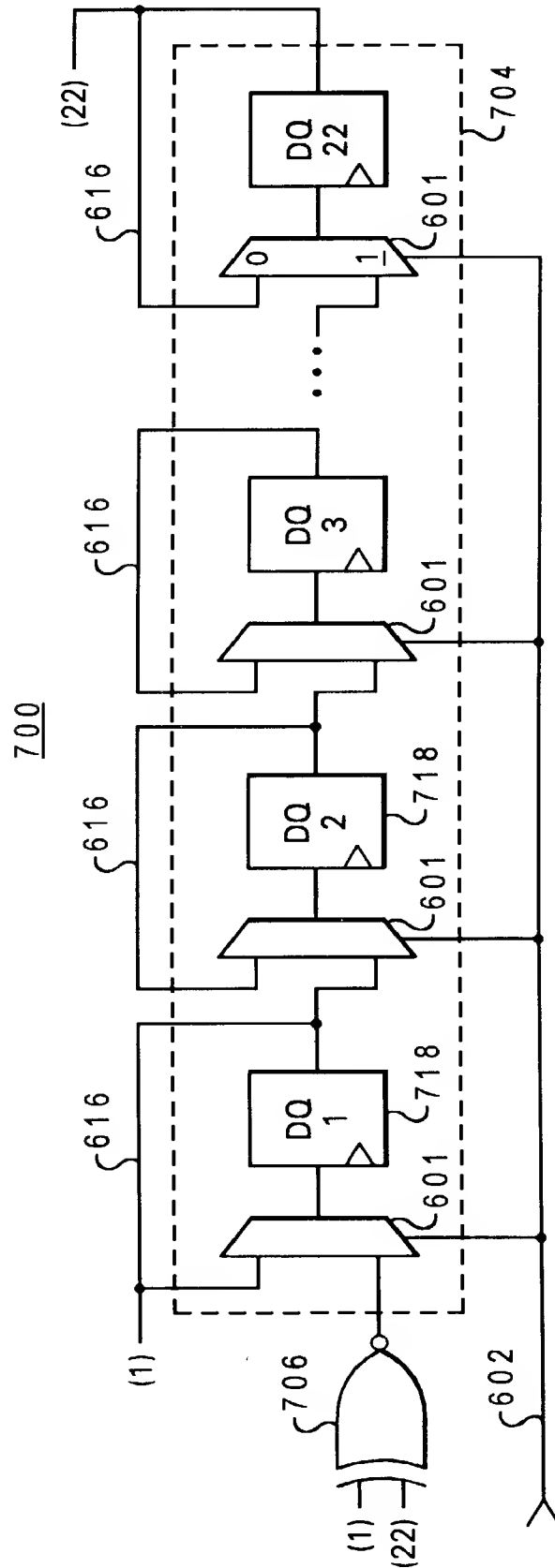


Fig. 7

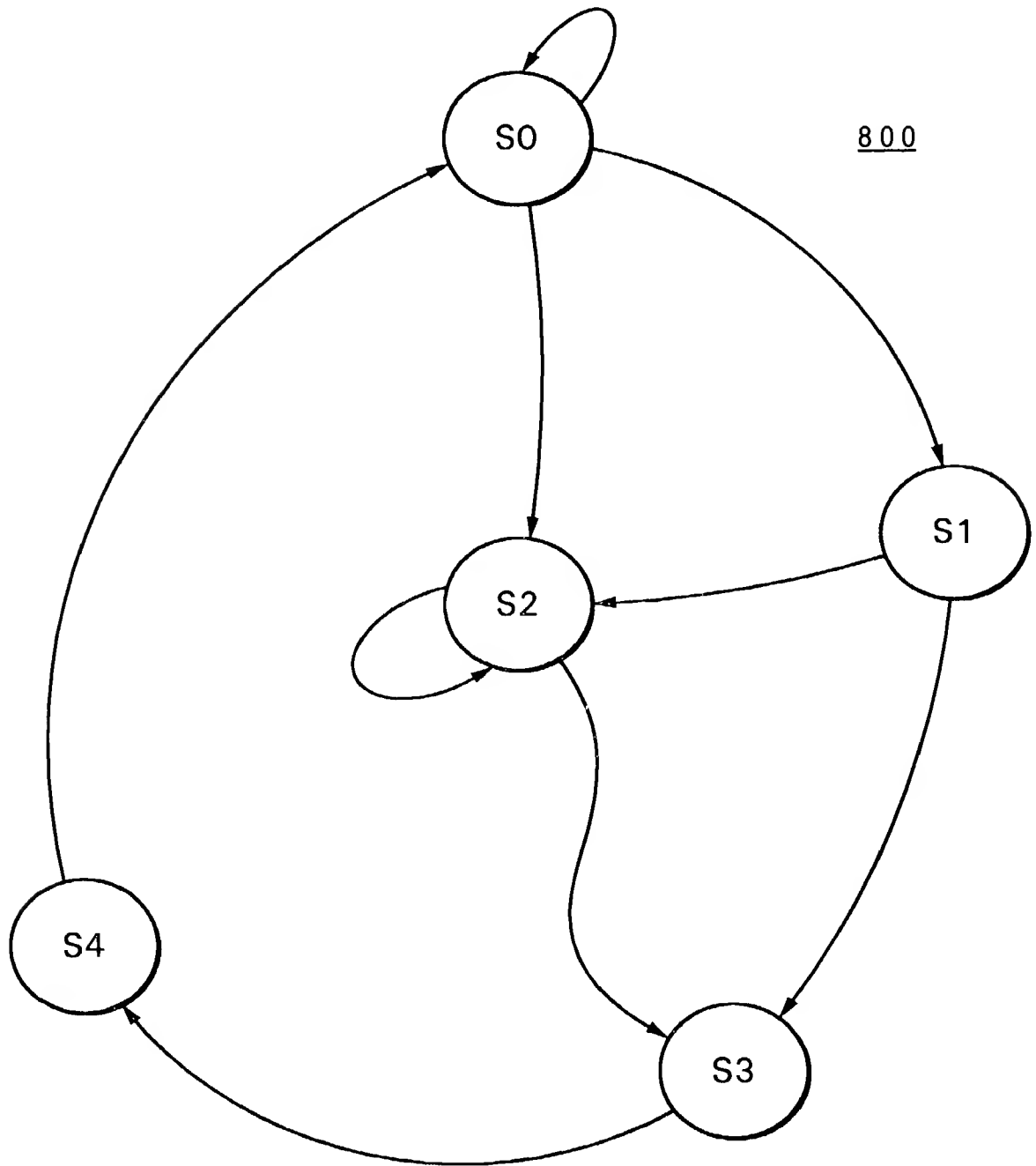


Fig. 8A
Prior Art

entity FSM : FSM

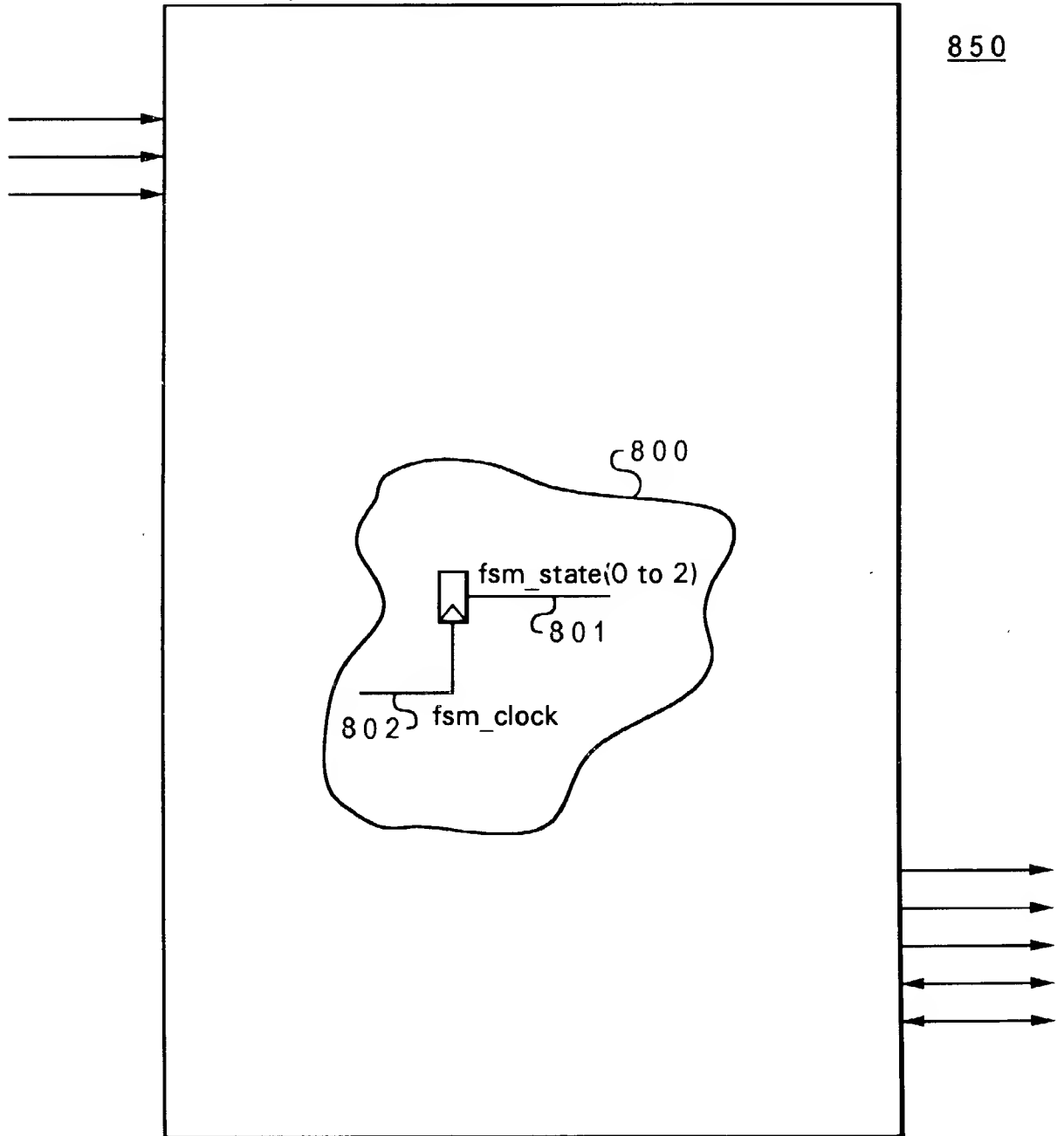


Fig. 8B

Prior Art

ENTITY FSM IS

PORT(
 ports for entity fsm....
);

ARCHITECTURE FSM OF FSM IS

BEGIN

 ... HDL code for FSM and rest of the entity ...

 fsm_state(0 to 2) <= ... Signal 801 ...

853	{	--!! Embedded FSM : examplefsm;	}	852	}	860
859	{	--!! clock : (fsm_clock);				
854	{	--!! state_vector : (fsm_state(0 to 2));				
855	{	--!! states : (S0, S1, S2, S3, S4);				
856	{	--!! state_encoding : ('000', '001', '010', '011', '100');				
857	{	--!! arcs : (S0 => S0, S0 => S1, S0 => S2,				
		--!! (S1 => S2, S1 => S3, S2 => S2,				
		--!! (S2 => S3, S3 => S4, S4 => S0);				
858	{	--!! End FSM;				

END;

Fig. 8C

20/62

entity FSM : FSM

850

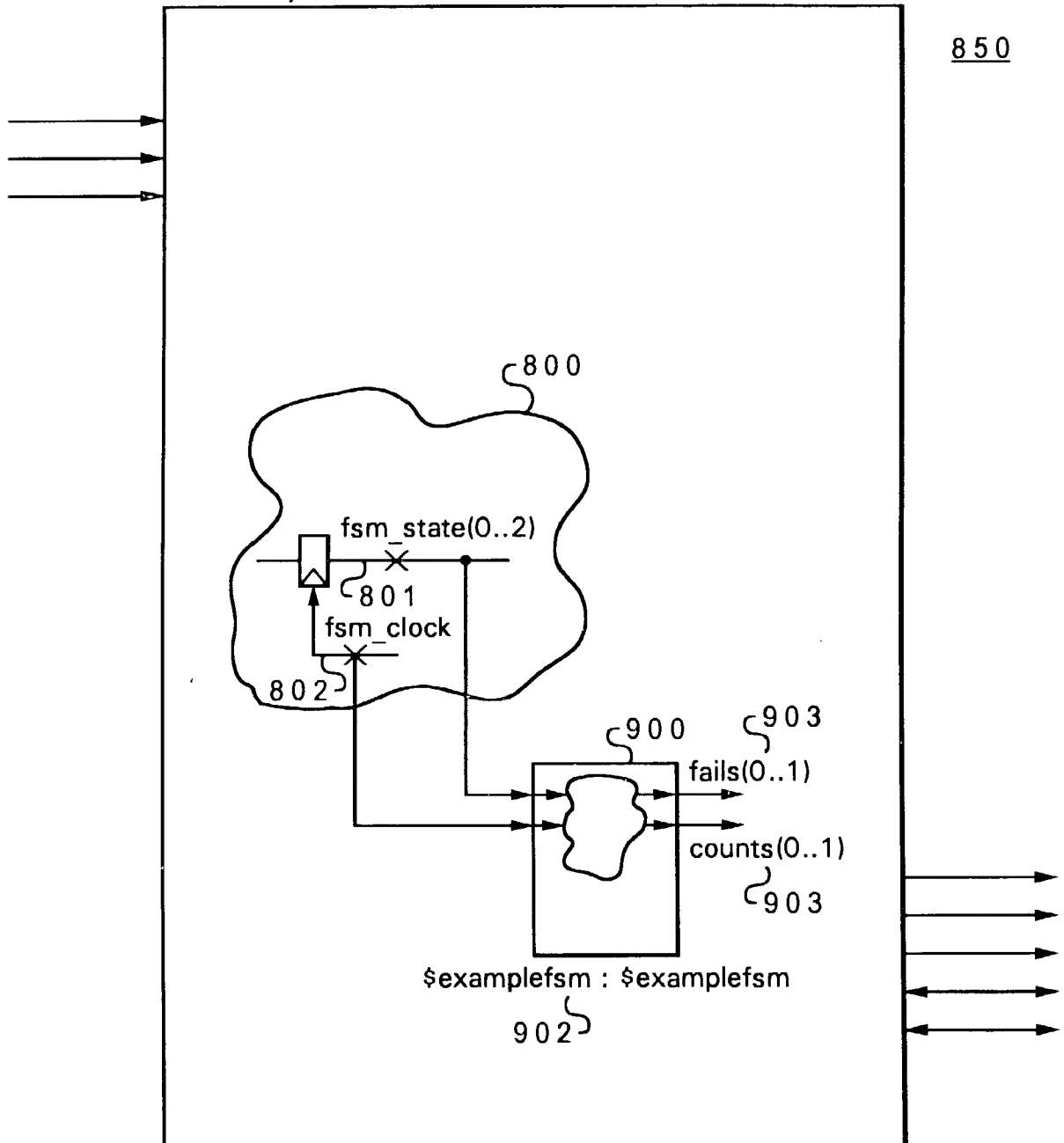
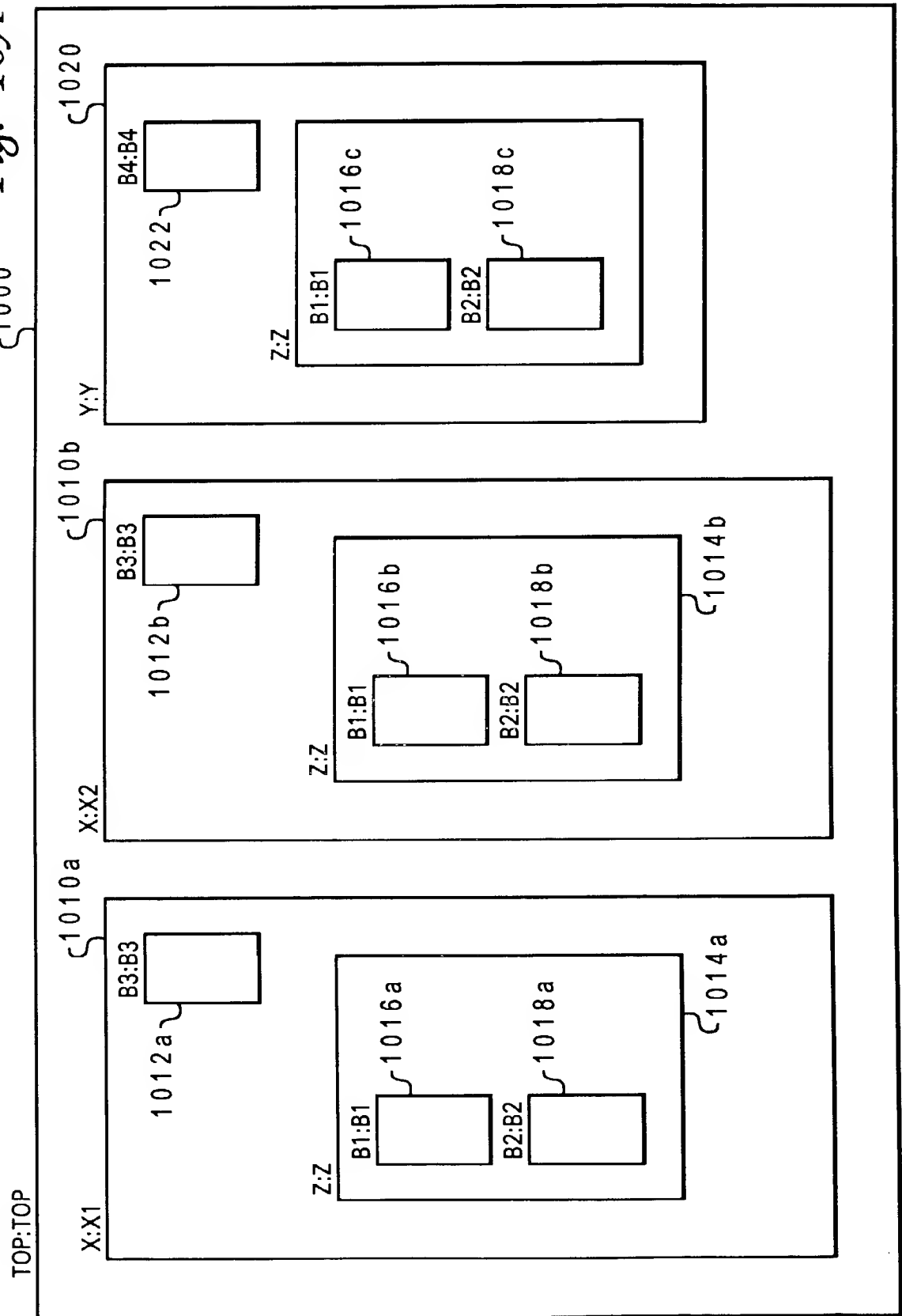


Fig. 9

Fig. 10A



1030 1032 1034 1036
<instantiation identifier> . <instrumentation entity name> . <design entity name> . <eventname>

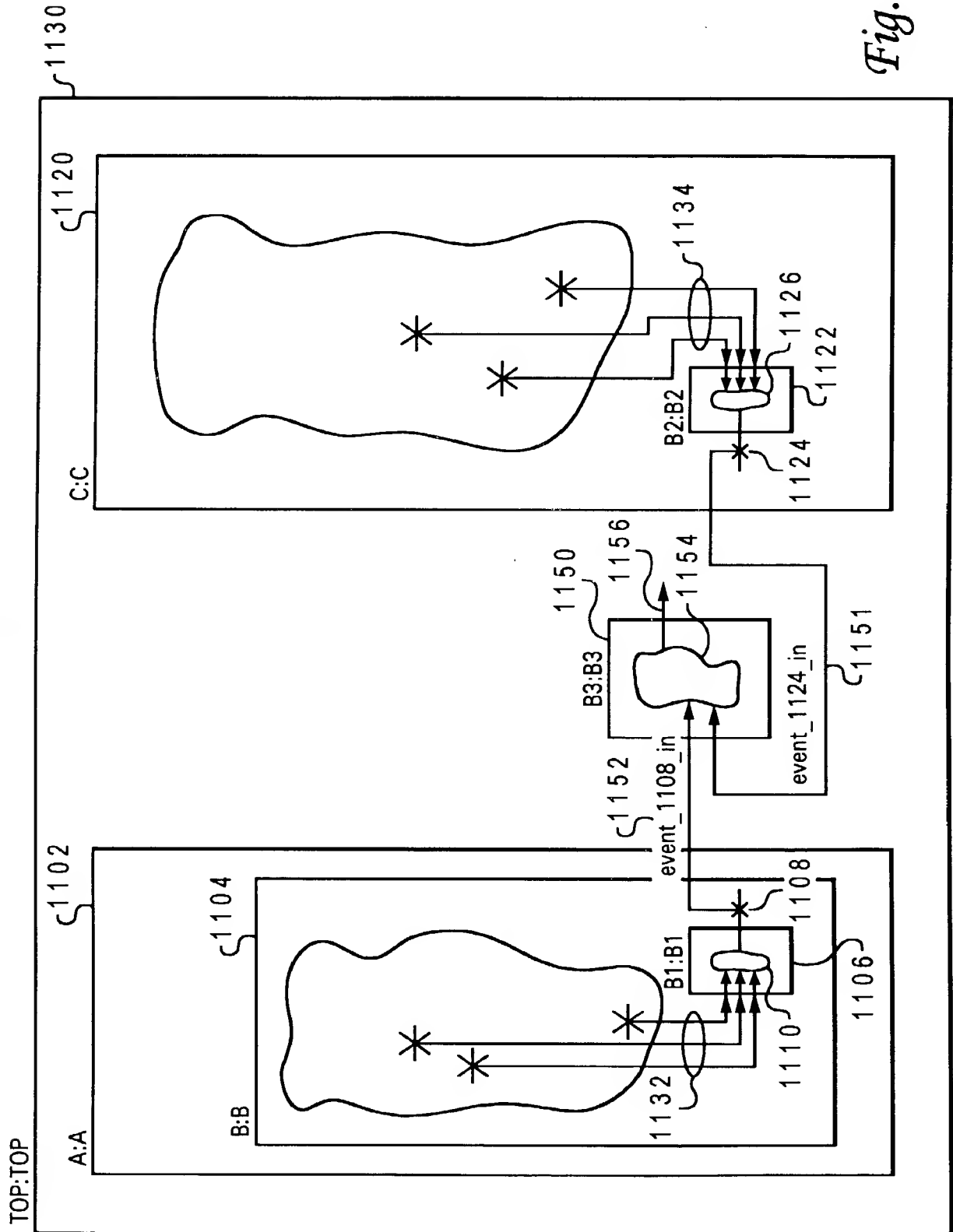
Fig. 10B

1030 1032 1034 1036
X1 B3 X COUNT1 1040
X1.Z B1 Z COUNT1 1041
X1.Z B2 Z COUNT1 1042
X2 B3 Z X COUNT1 1043
X2.Z B1 Z COUNT1 1044
X2.Z B2 Z COUNT1 1045
Y B4 Y COUNT1 1046
Y.Z B1 Z COUNT1 1047
Y.Z B2 Z COUNT1 1048

Fig. 10C

1030 1034 1036
<instantiation identifier> . <design entity name> . <eventname>

Fig. 10D



24/62

--!! Inputs
 --!! event_1108_in <= C.[B2.count.event_1108];
 --!! event_1124_in <= A.B.[B1.count.event_1124];
 --!! End Inputs

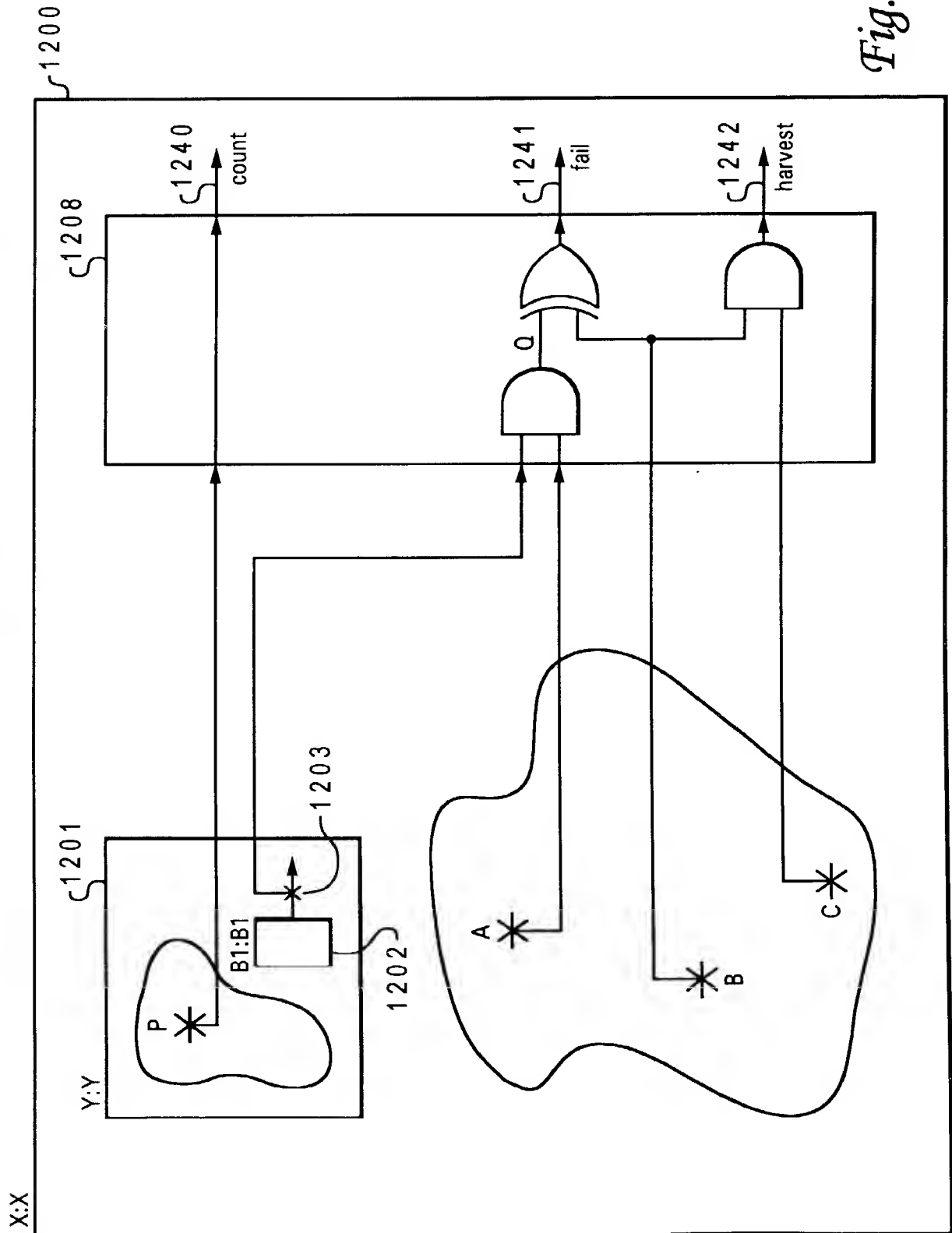
1163 } 1165 } 1161 }
 1164 } 1166 } 1162 }

Fig. 11B

--!! Inputs
 --!! event_1108_in <= C.[count.event_1108];
 --!! event_1124_in <= B.[count.event_1124];
 --!! End Inputs

1171 }
 1172 }

Fig. 11C



ENTITY X IS

PORT(:
:
:
);

ARCHITECTURE example of X IS

BEGIN

.
.
.
.
... HDL code for X ...
.
.
.
.

1220

1221 { Y:Y
PORT MAP(:
:
);

1222 { A <=
B <=
C <=

1223 { --!! [count, countname0, clock] <= Y.P; 1230
--!! Q <= Y. [B1.count.count1] AND A; 1232
--!! [fail, failname0, "fail msg"] <= Q XOR B; 1234
--!! [harvest, harvestname0, "harvest msg"] <= B AND C;
END; 1236

Fig. 12B

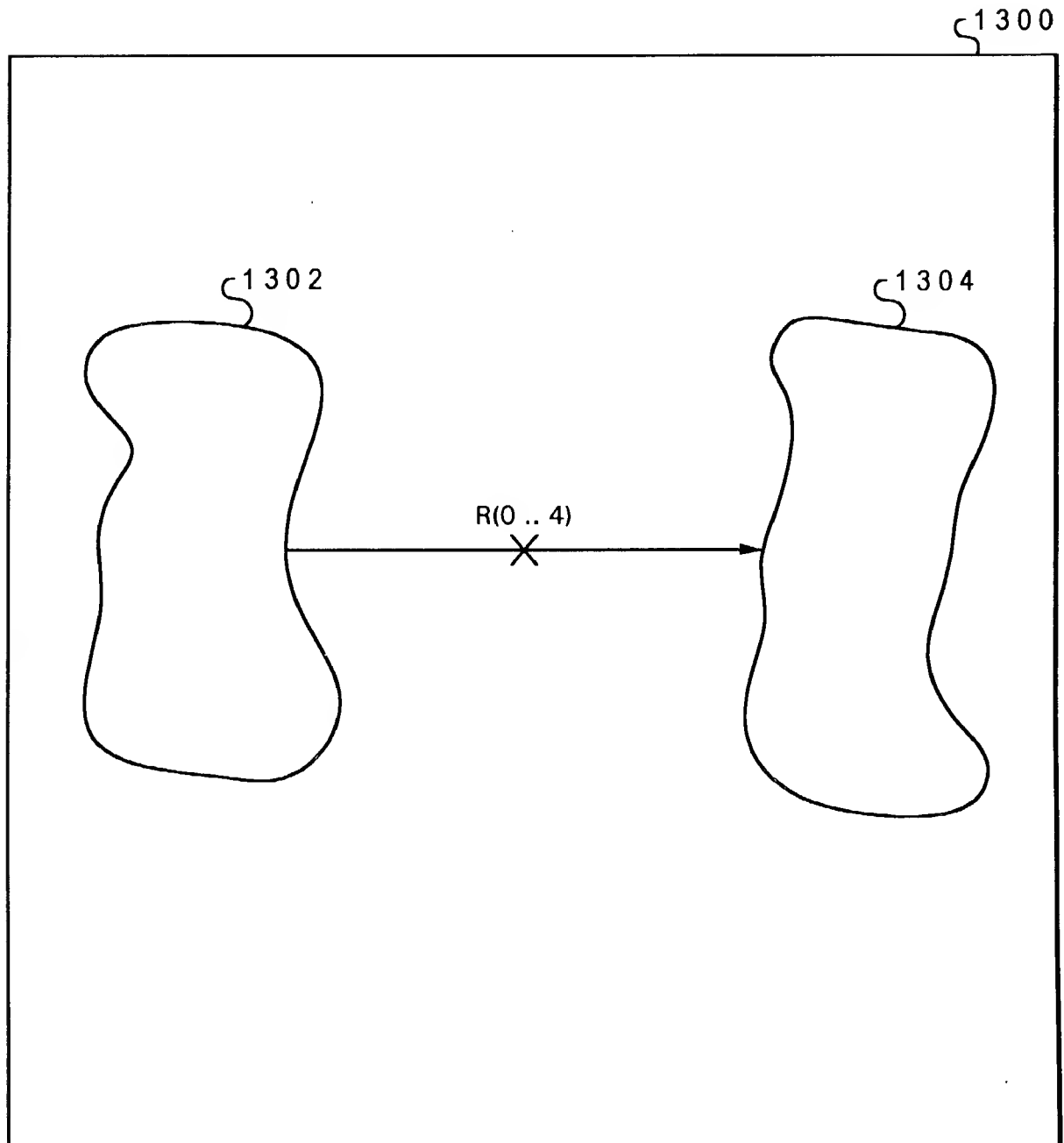


Fig. 13A

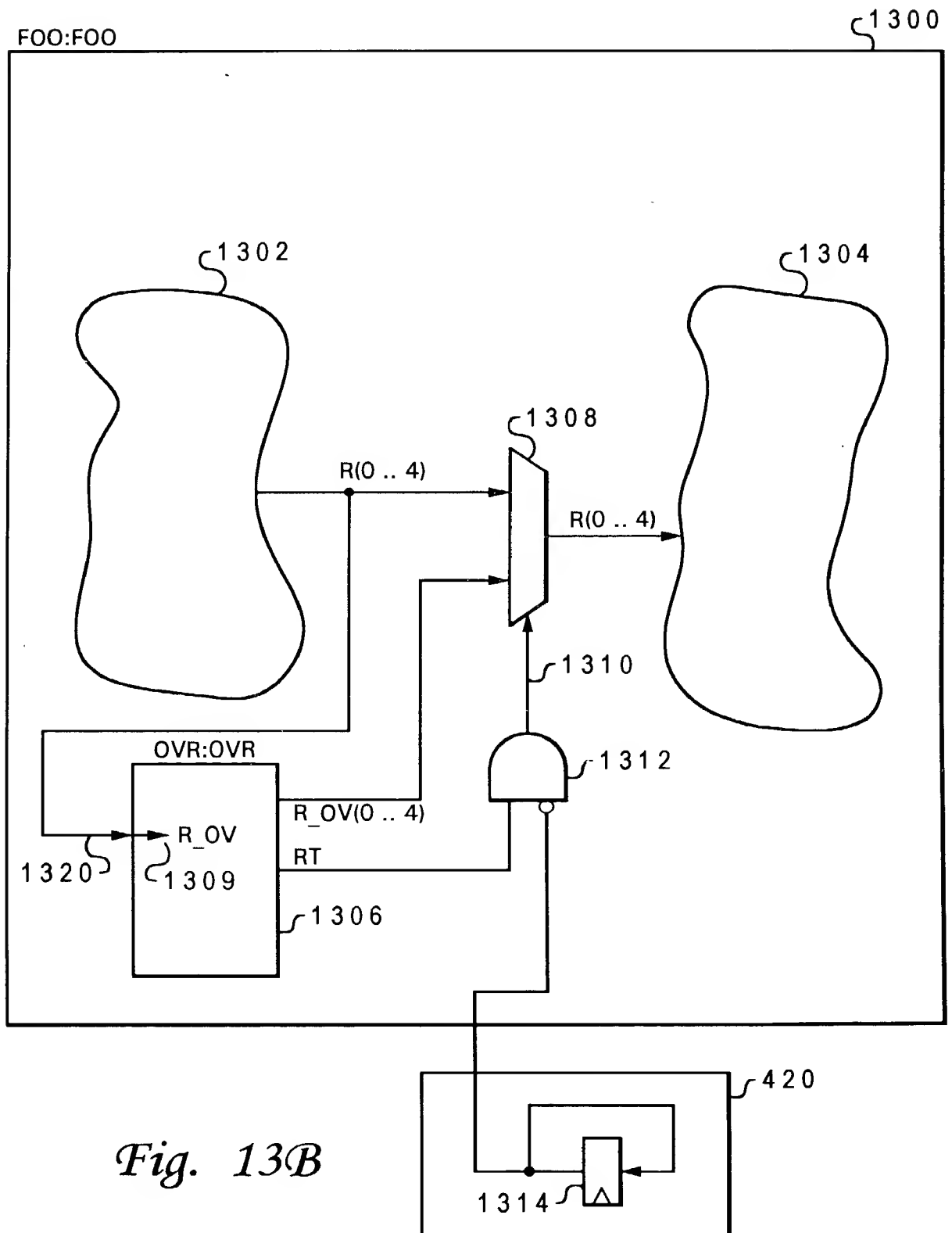


Fig. 13B

29/62

```

ENTITY OVR IS
    PORT(  R_IN      :  IN std_ulogic_vector(0 .. 4);
           .
           .
           ... other ports as required ...
           .
           .
           R_OV      :  OUT std_ulogic_vector(0 .. 4);
           RT        :  OUT std_ulogic
    );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN => {R(0 .. 4)};
--!! :
--!! ... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS
    BEGIN
        ... HDL code for entity body section ...
    END;
    
```

Handwritten annotations in the figure include curly braces and numbers grouping lines of code:

- A brace labeled **1 3 6 4** groups the `IN std_ulogic_vector(0 .. 4);` line.
- A brace labeled **1 3 6 2** groups the `OUT std_ulogic_vector(0 .. 4);` line.
- A brace labeled **1 3 6 3** groups the `OUT std_ulogic` line.
- A brace labeled **1 3 6 0** groups the `--!! R_IN => {R(0 .. 4)};` line.
- A brace labeled **1 3 6 1** groups the `--!! <R_OVRIDE> : R_OV(0 .. 4) => R(0 .. 4) [RT];` line.
- A brace labeled **1 3 5 6** groups the entire `--!!` comment block.
- A brace labeled **1 3 5 1** groups the `--!!` comment block.
- A brace labeled **1 3 5 8** groups the `... HDL code for entity body section ...` line.
- A large brace labeled **1 3 4 0** groups the entire `ENTITY OVR IS` block.

Fig. 13C

ENTITY FOO IS

PORT(:
:
:
);

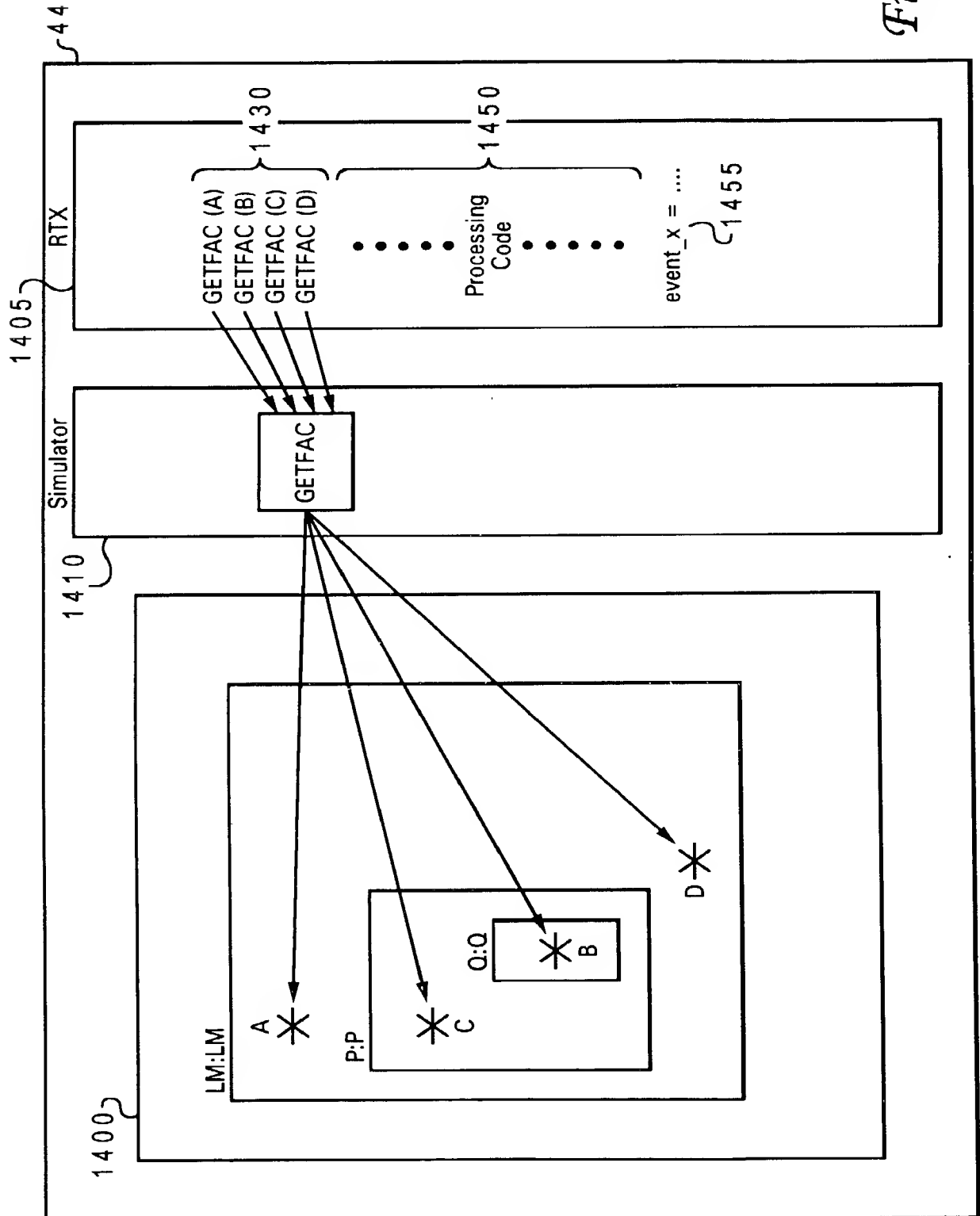
ARCHITECTURE example of FOO IS

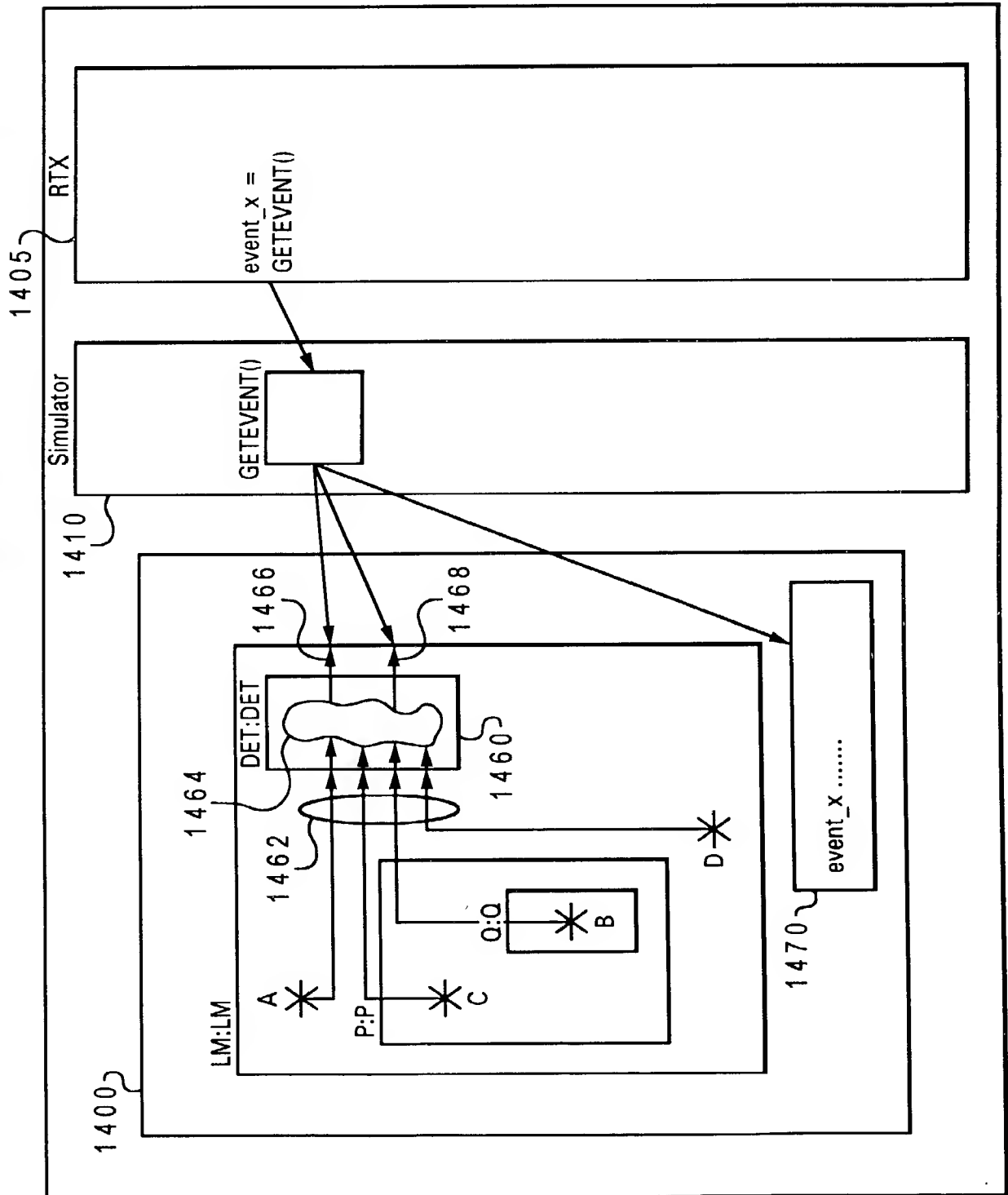
BEGIN

.
.
.
.
.
R <=
.
.
.
.

1380 { --!! R_IN <= {R}; 1381
--!! 1382
--!! R_OV(0 to 4) <=; 1383
--!! RT <=;
--!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4); 1384

Fig. 13D






```

ENTITY DET IS
    PORT(
        A      : IN std_ulogic;
        B      : IN std_ulogic_vector(0 to 5);
        C      : IN std_ulogic;
        D      : IN std_ulogic;
        :
        :
        event_x : OUT std_ulogic_vector(0 to 2);
        x_here  : OUT std_ulogic;
    );

    --!! BEGIN
    --!! Design Entity: LM;

    --!! Inputs
    --!! A  => A;
    --!! B  => P.Q.B;
    --!! C  => P.C;
    --!! D  => D;
    --!! End Inputs

    --!! Detections
    --!! <event_x>:event_x(0 to 2) [x_here];
    --!! End Detections

    --!! End;

    ARCHITECTURE example of DET IS
    BEGIN
        ... HDL code ...

    END;

```

1491 {

1493 {

1495 {

1494 {

1480 {

1492 {

Fig. 14C

1660

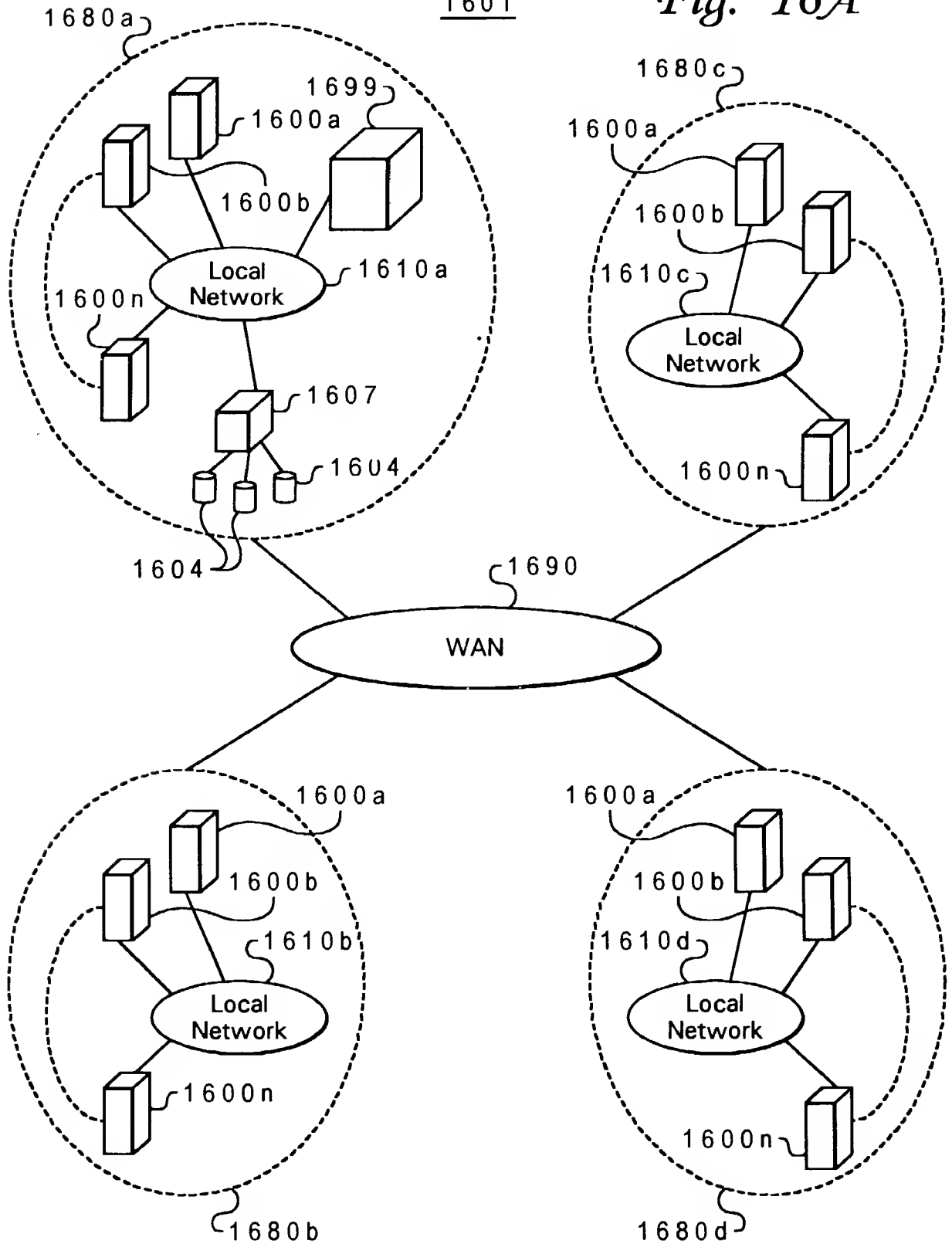
	1661		1662	
	1:	X1	B3	X
	2:	X1.Z	B1	Z
	3:	X1.Z	B2	Z
	4:	X2	B3	X
1663	5:	X2.Z	B1	Z
	6:	X2.Z	B2	Z
	7:	Y	B4	Y
	8:	Y.Z	B1	Z
	9:	Y.Z	B2	Z
				COUNT1

Fig. 15

35/62

1601

Fig. 16A



36/62

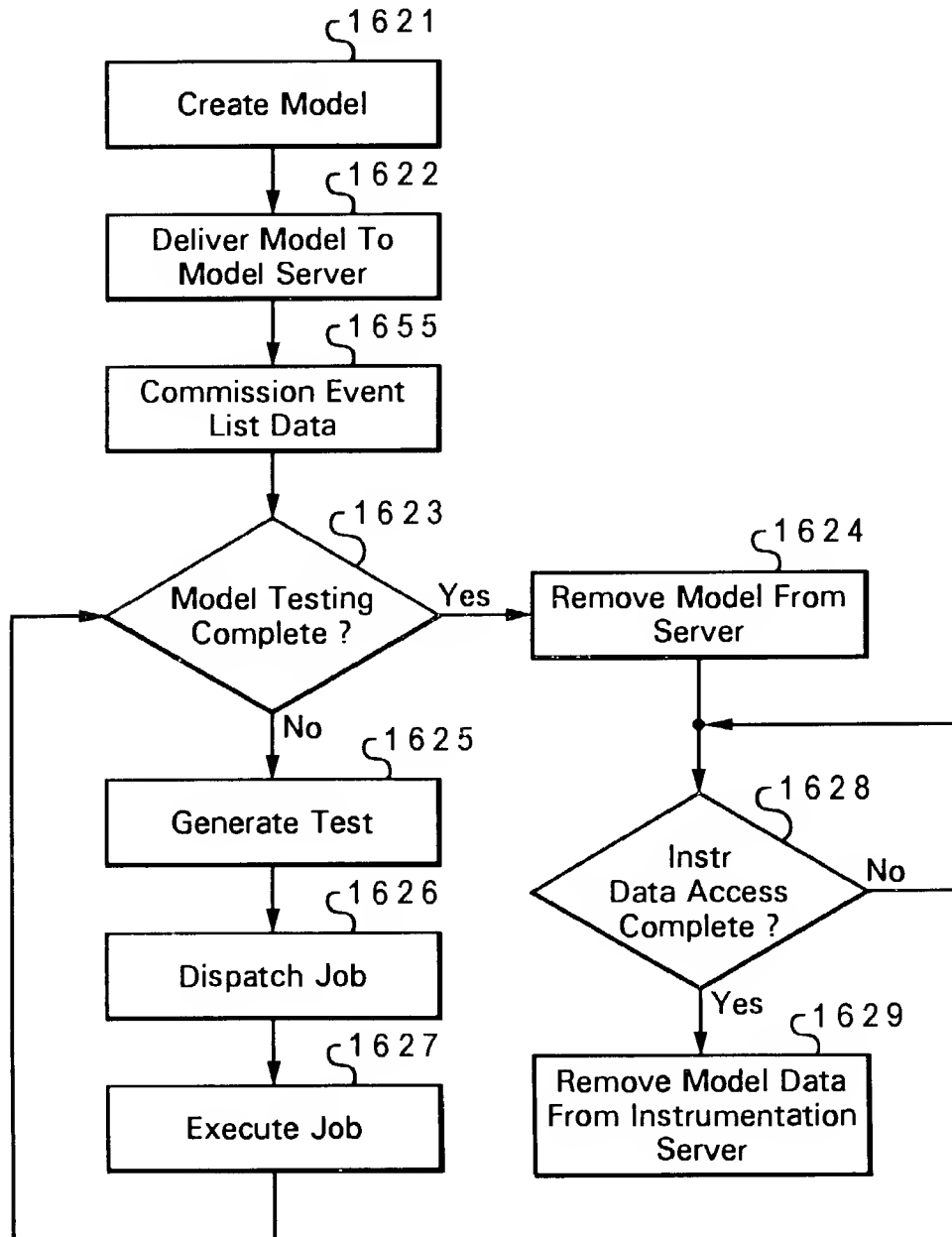


Fig. 16B

37/62

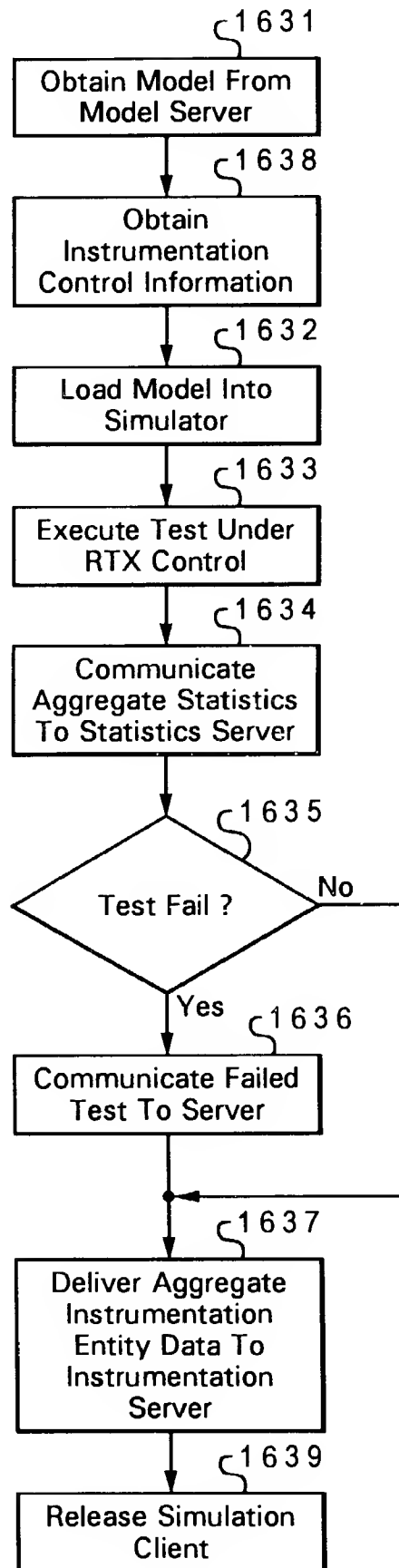


Fig. 16C

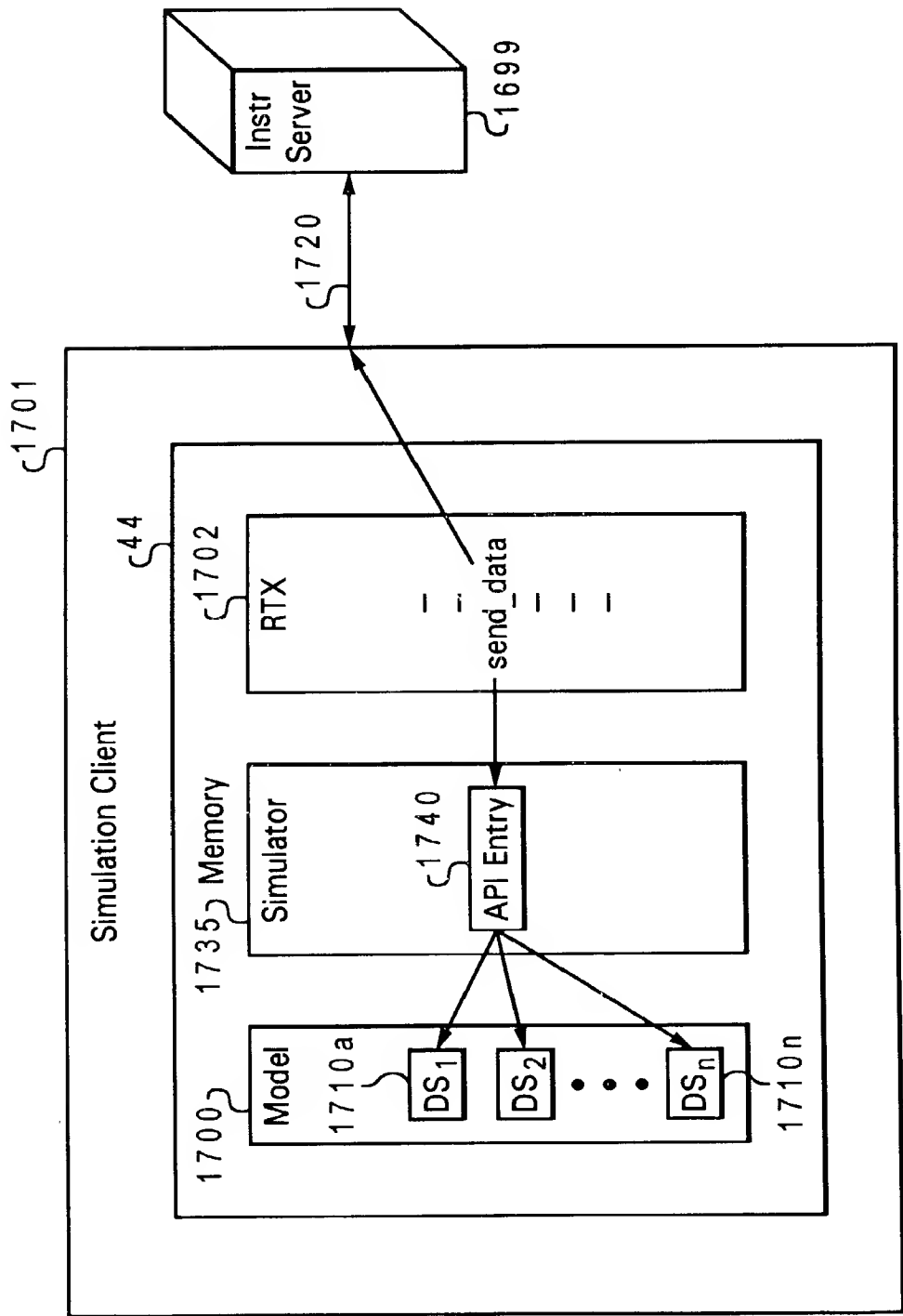


Fig. 17A

1750

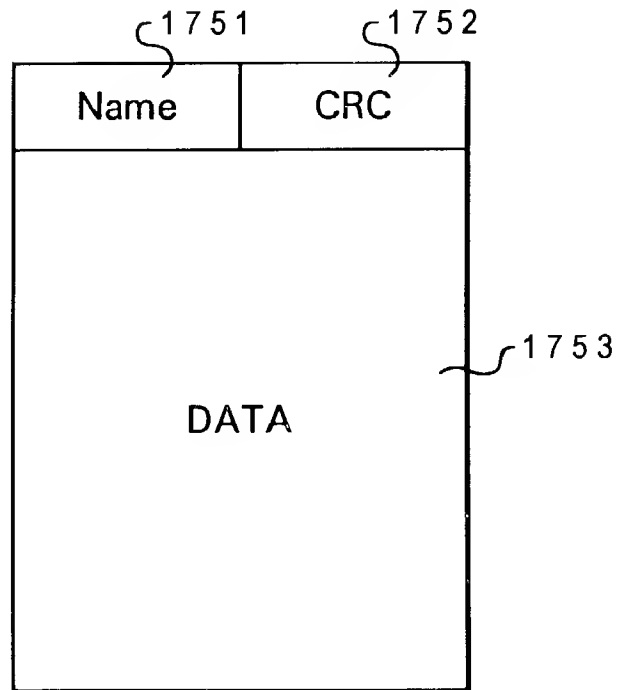


Fig. 17B

40/62

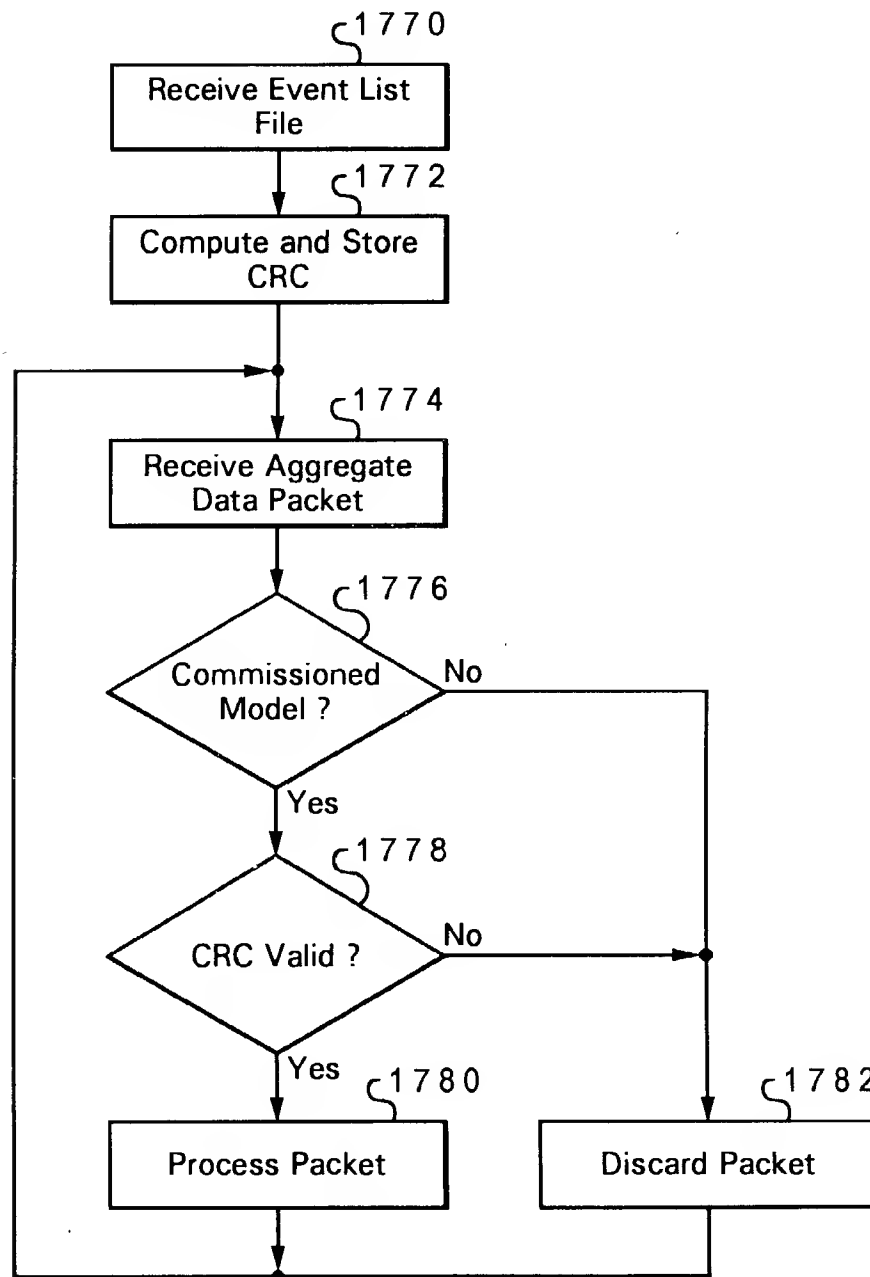


Fig. 17C

41/62

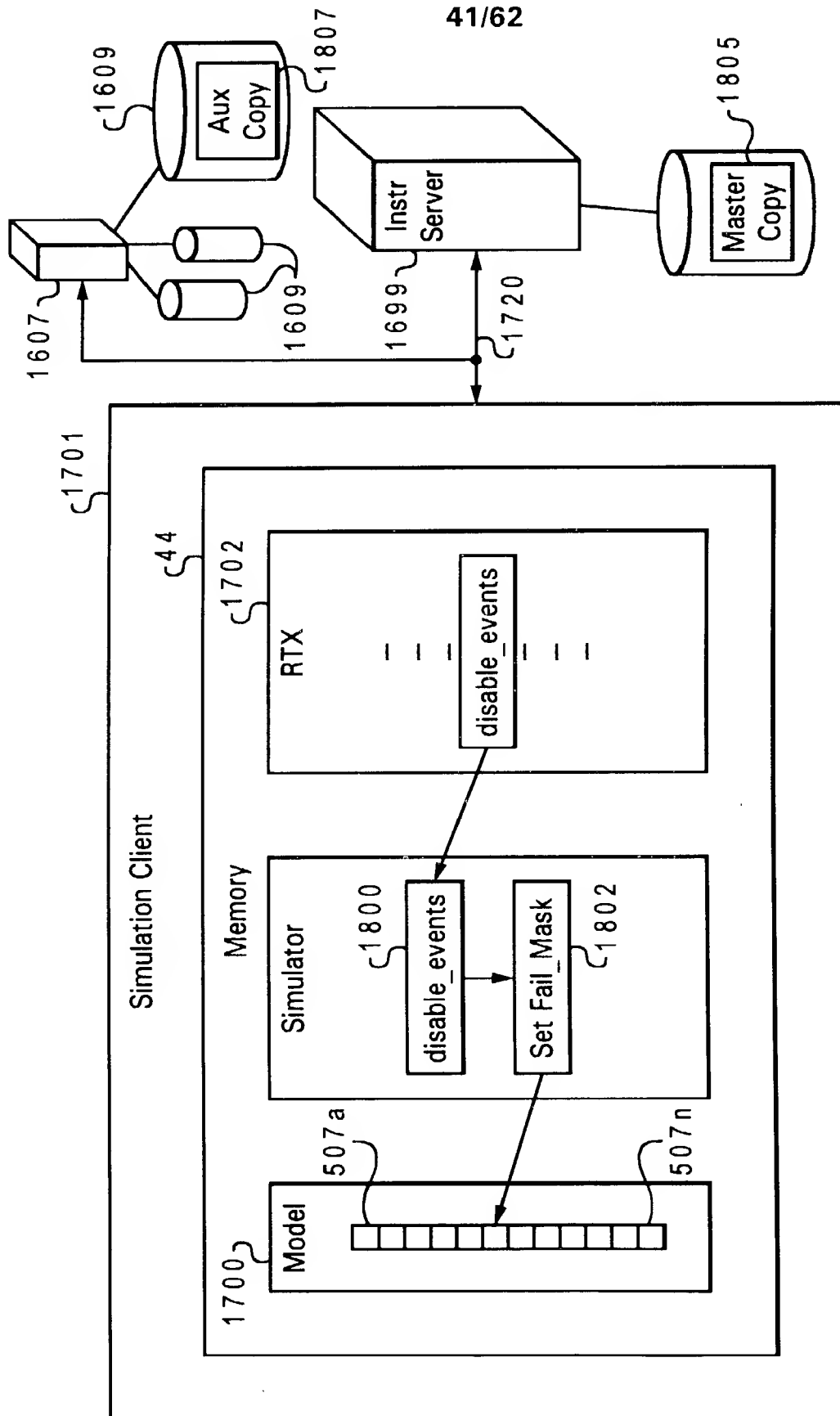
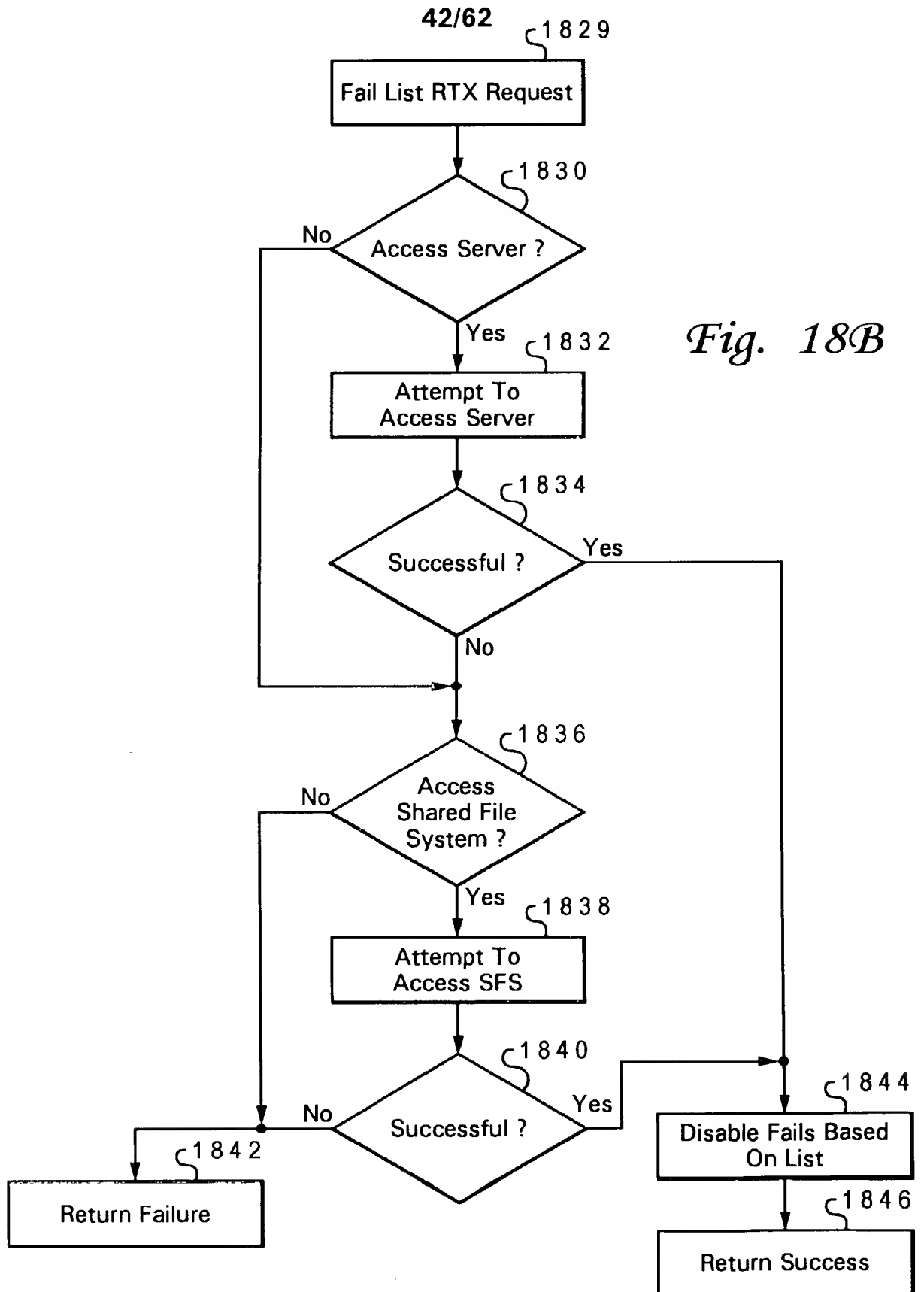


Fig. 18A



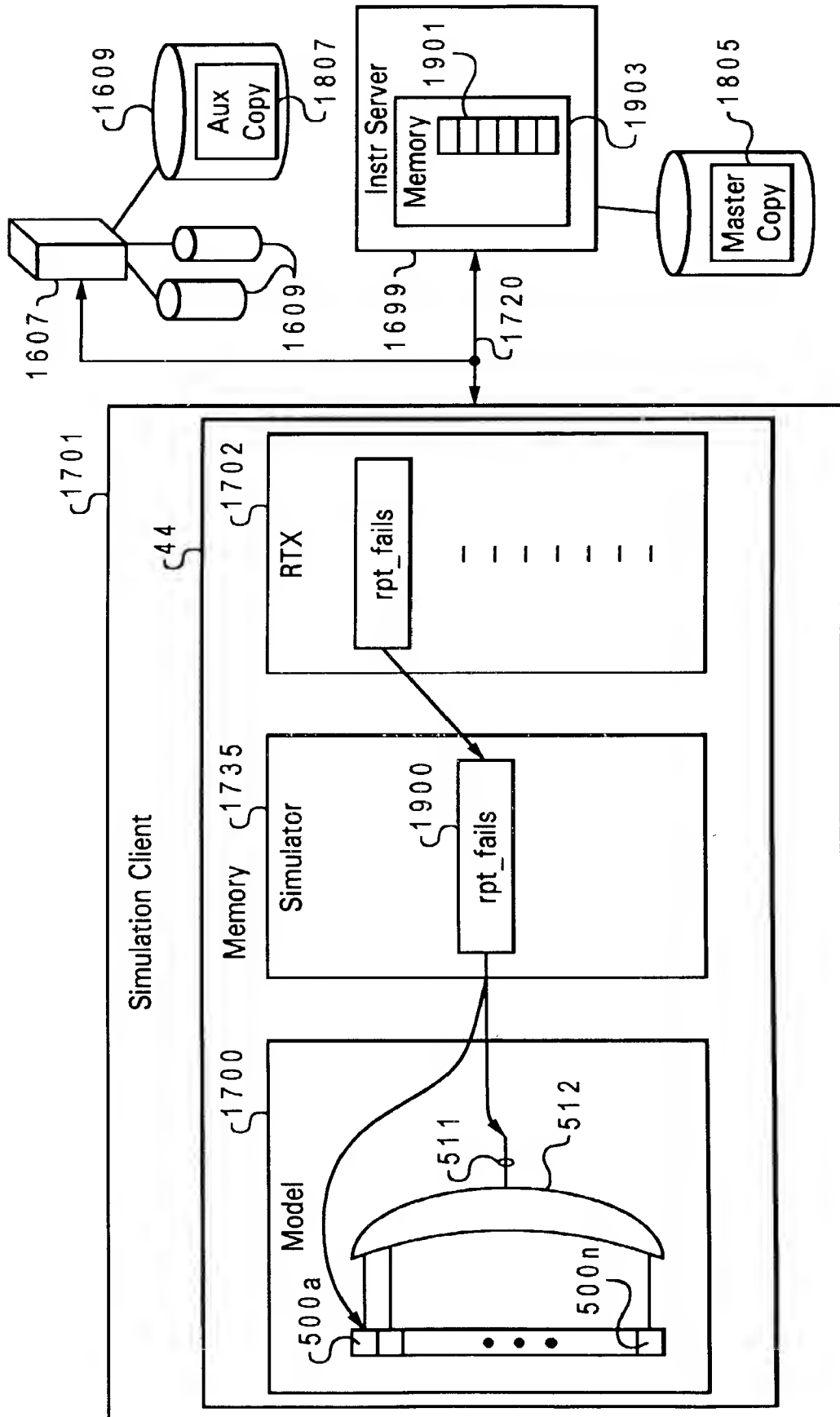
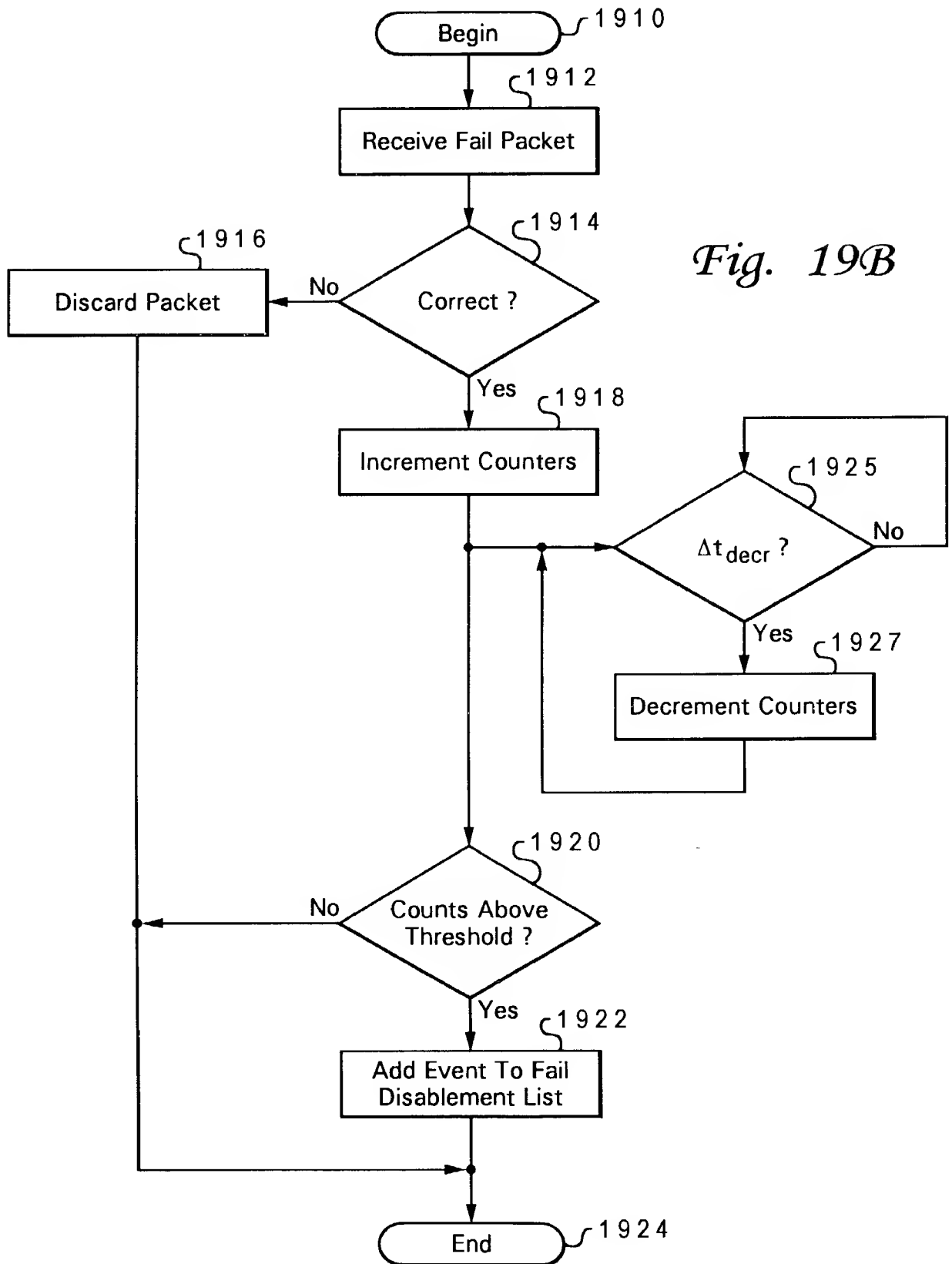


Fig. 19A

44/62



45/62

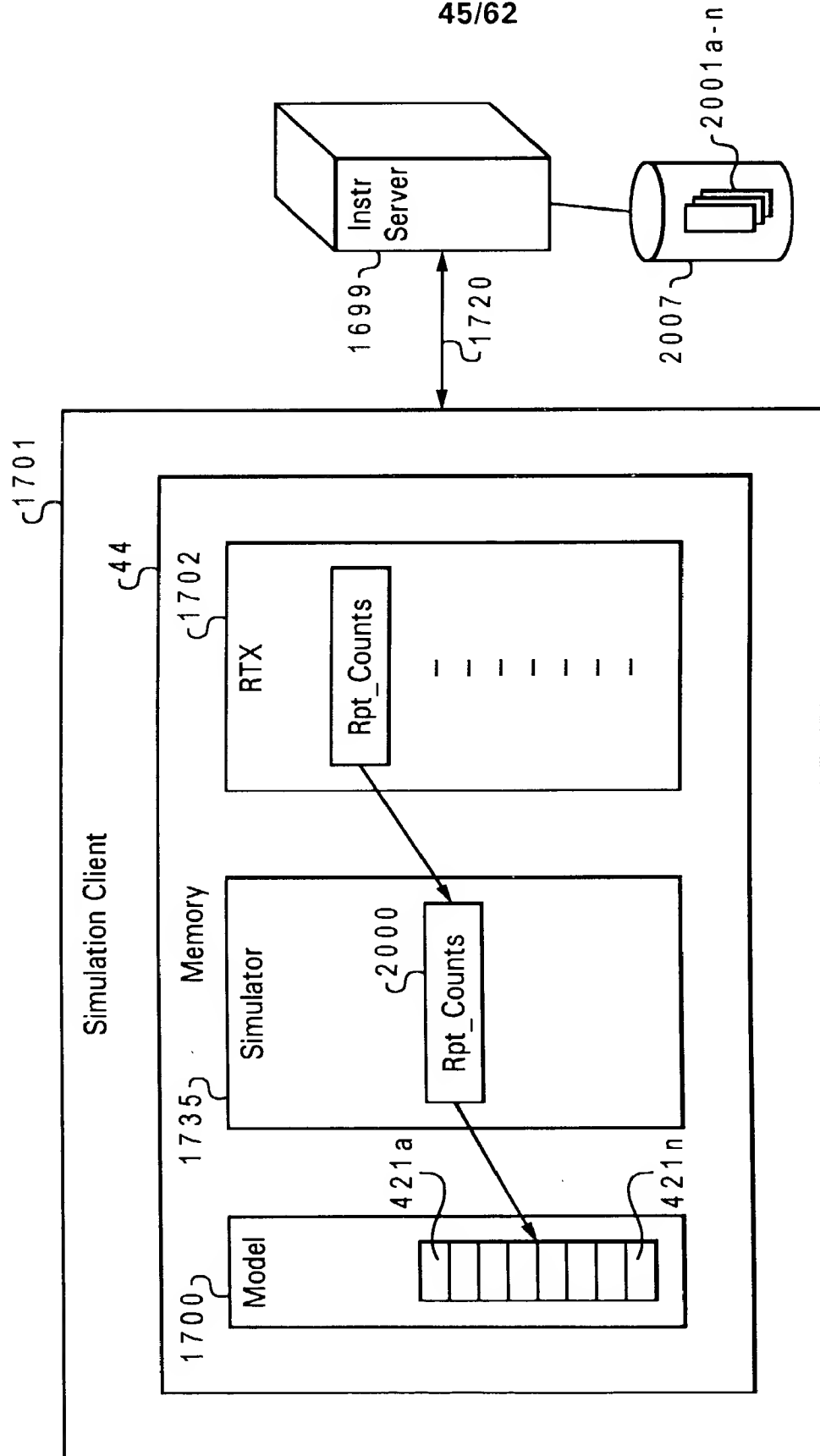


Fig. 20A

46/62

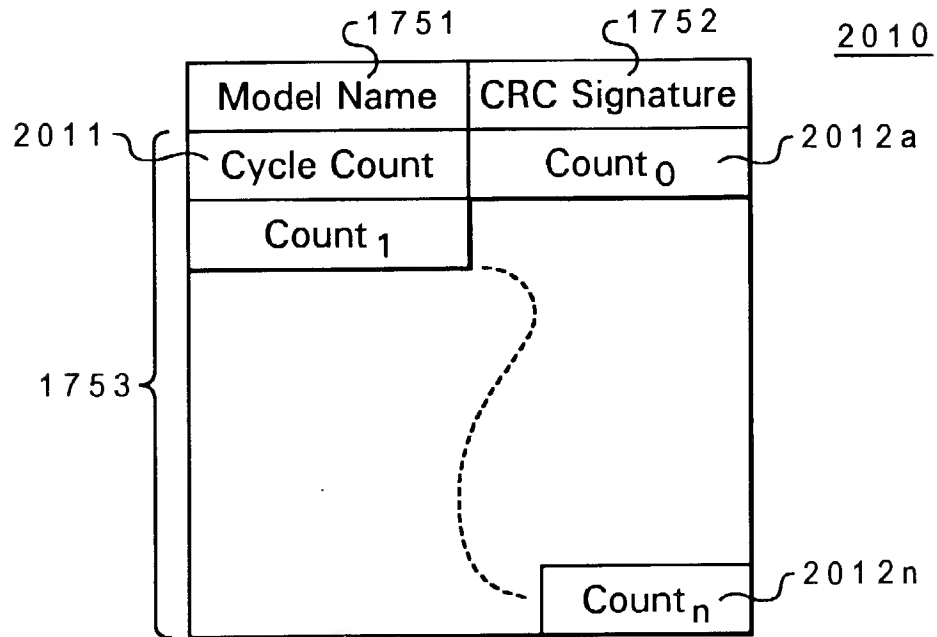


Fig. 20B

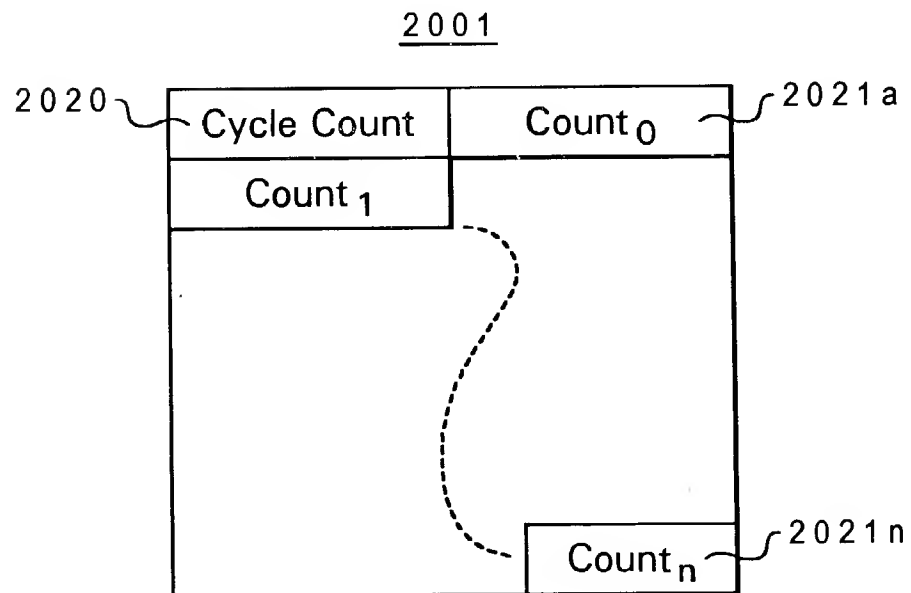


Fig. 20C

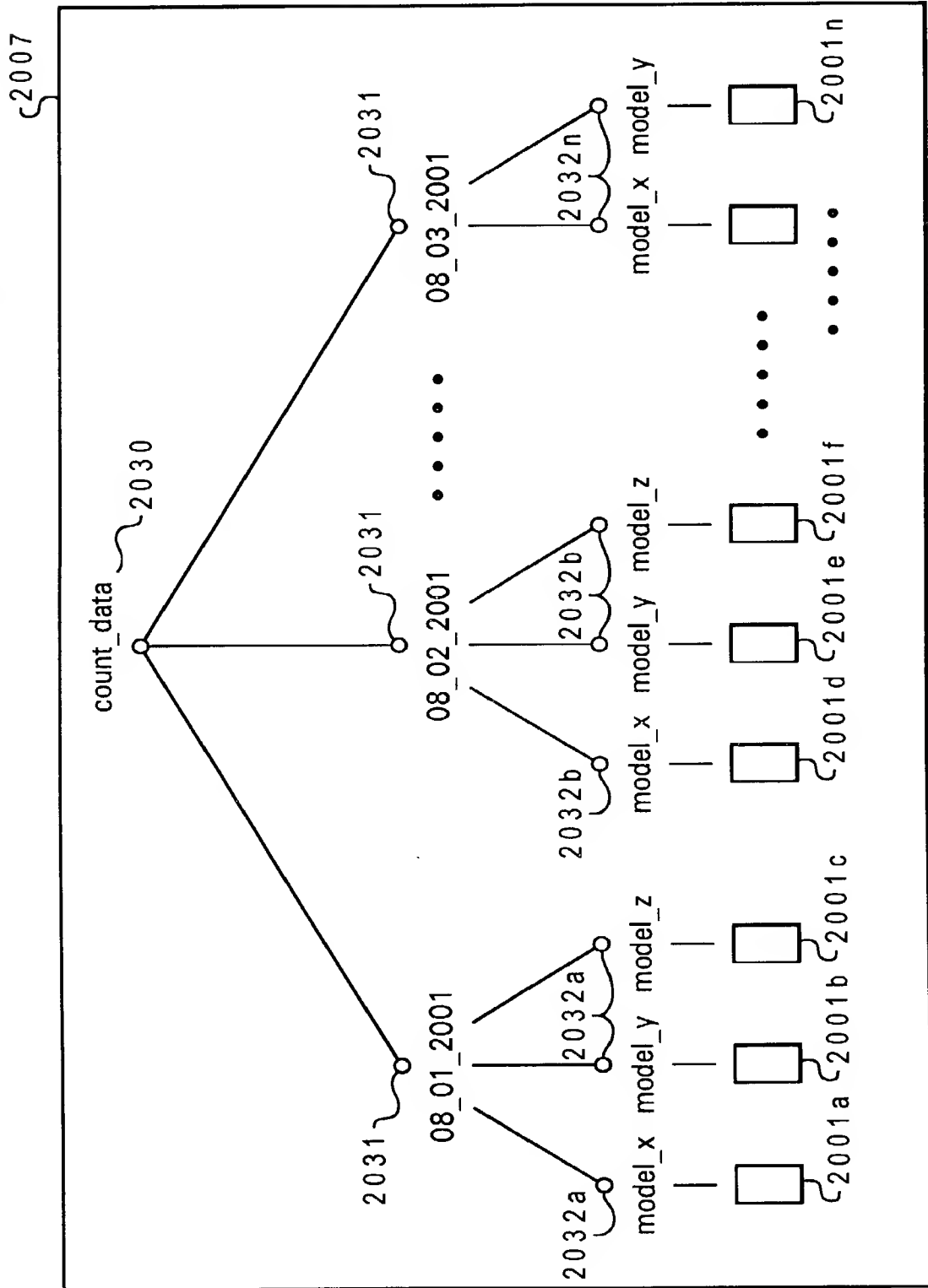


Fig. 20D

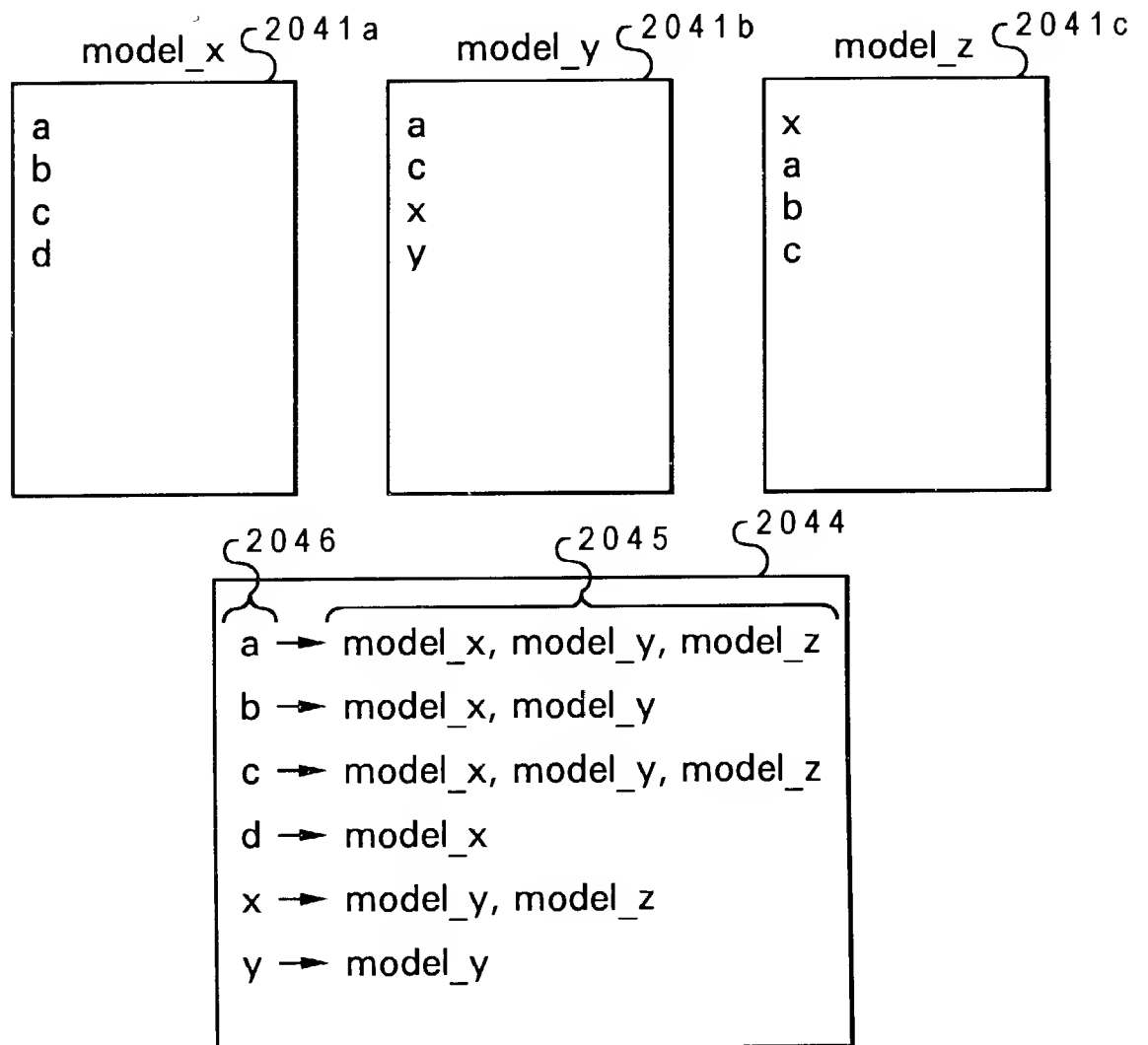


Fig. 20E

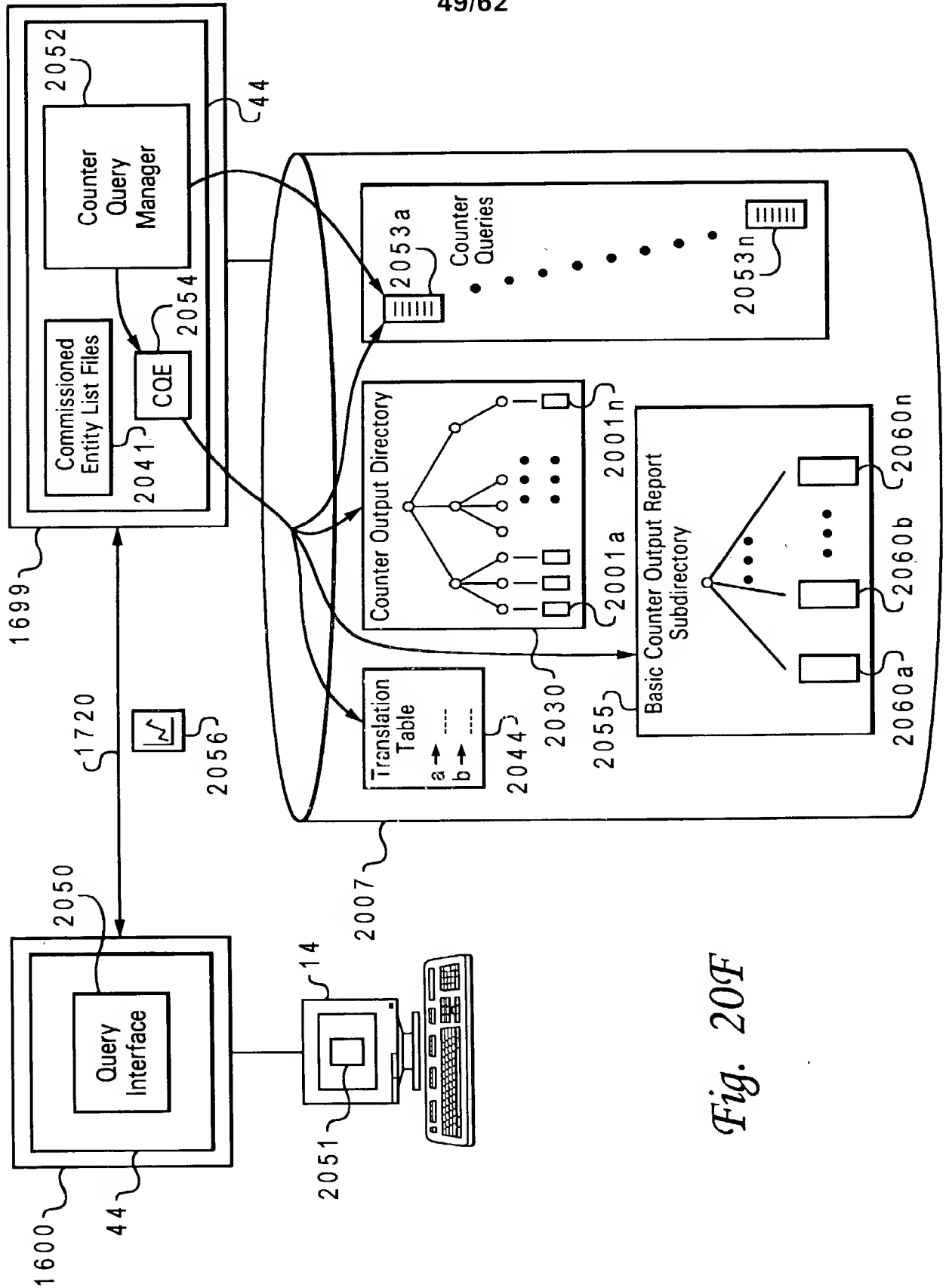
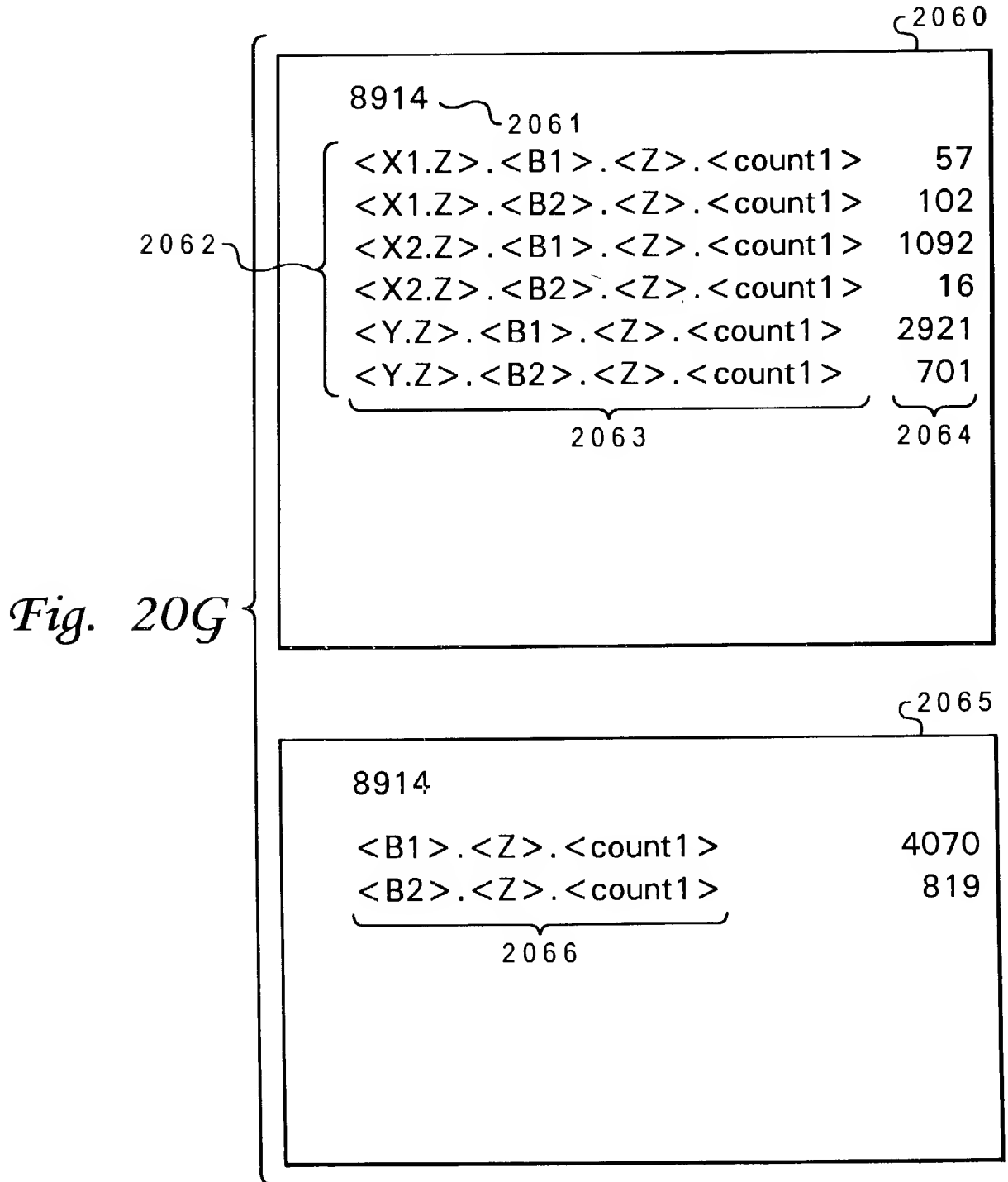


Fig. 20F



51/62

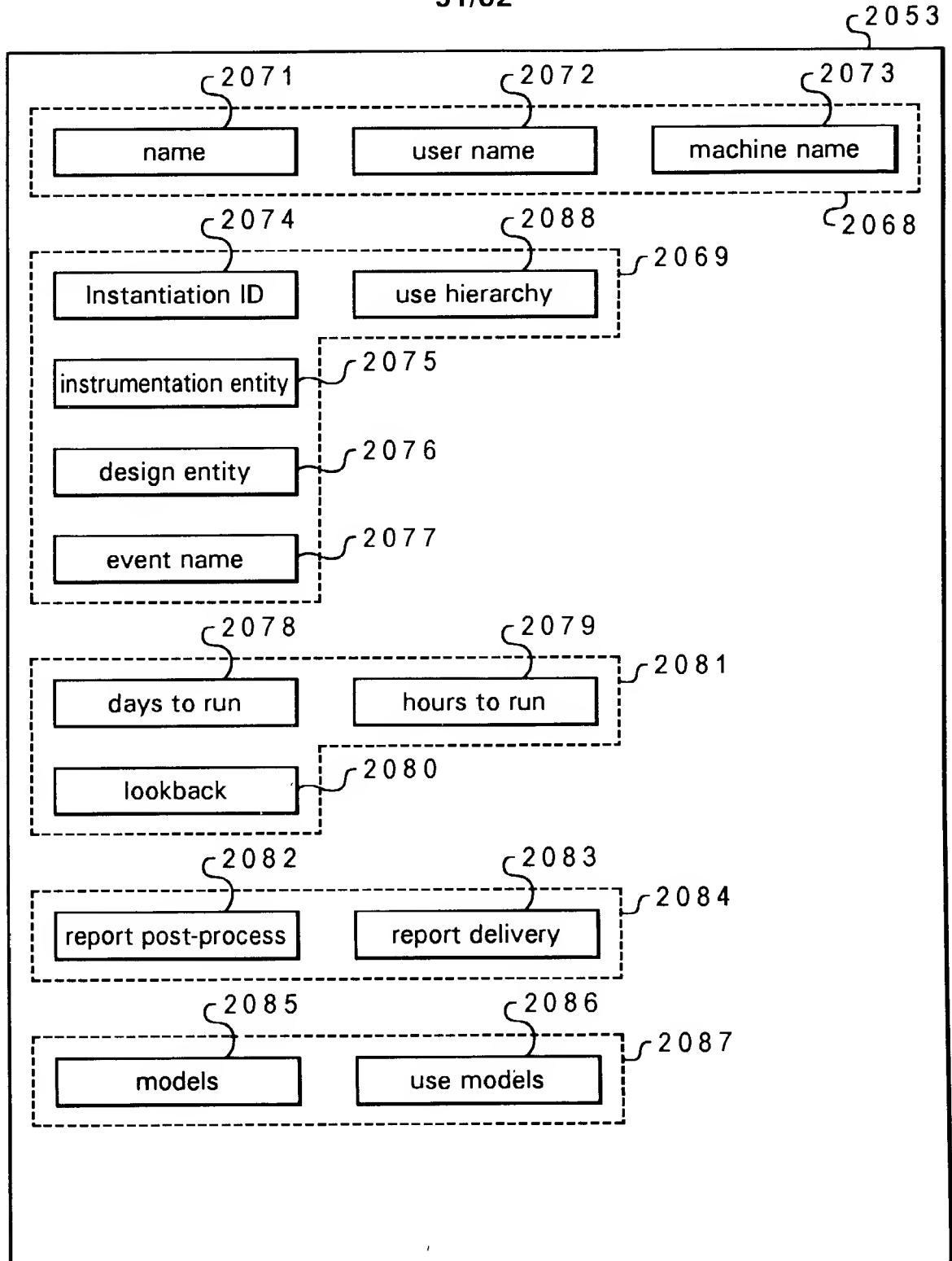


Fig. 20H

52/62

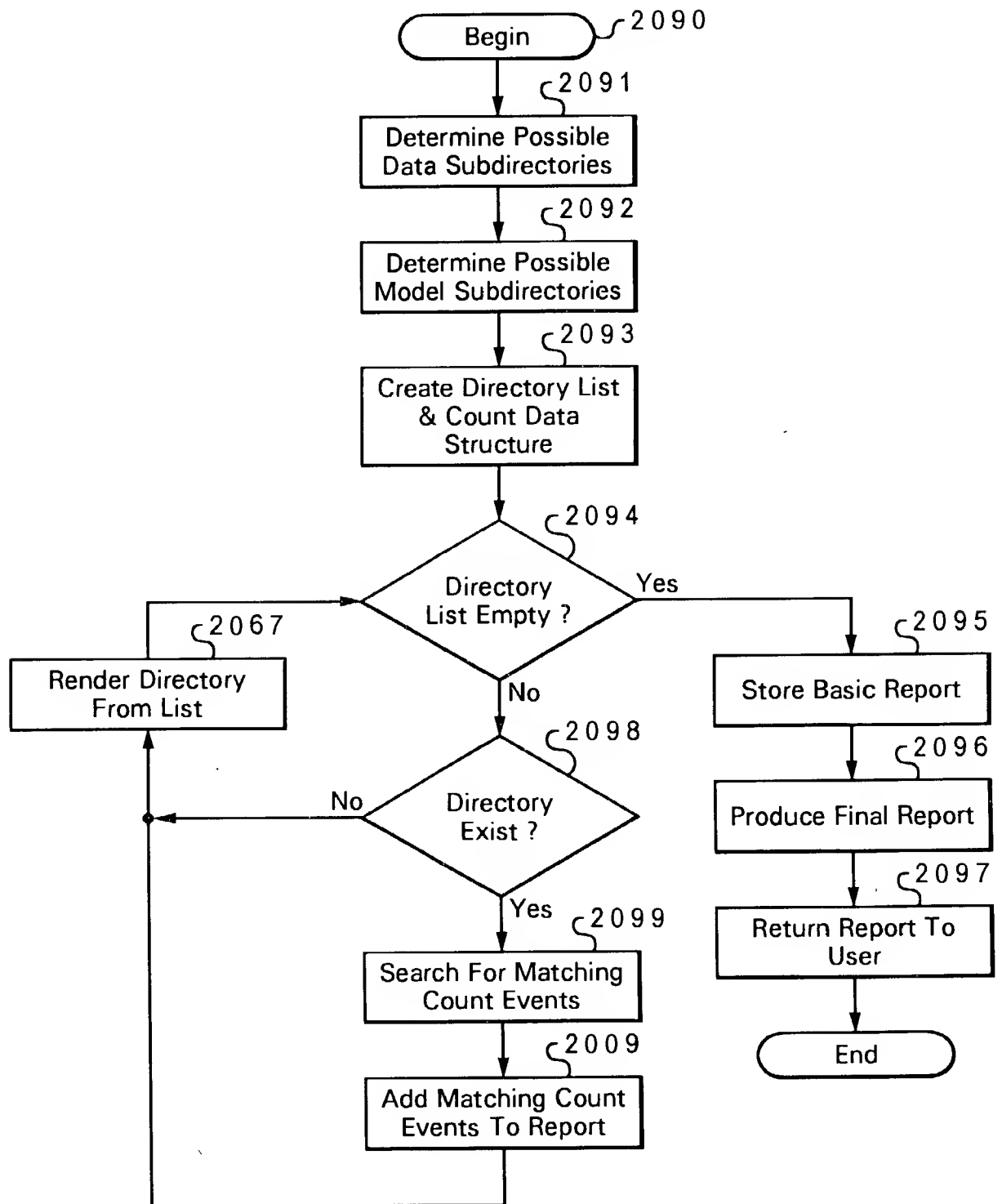


Fig. 20I

53/62

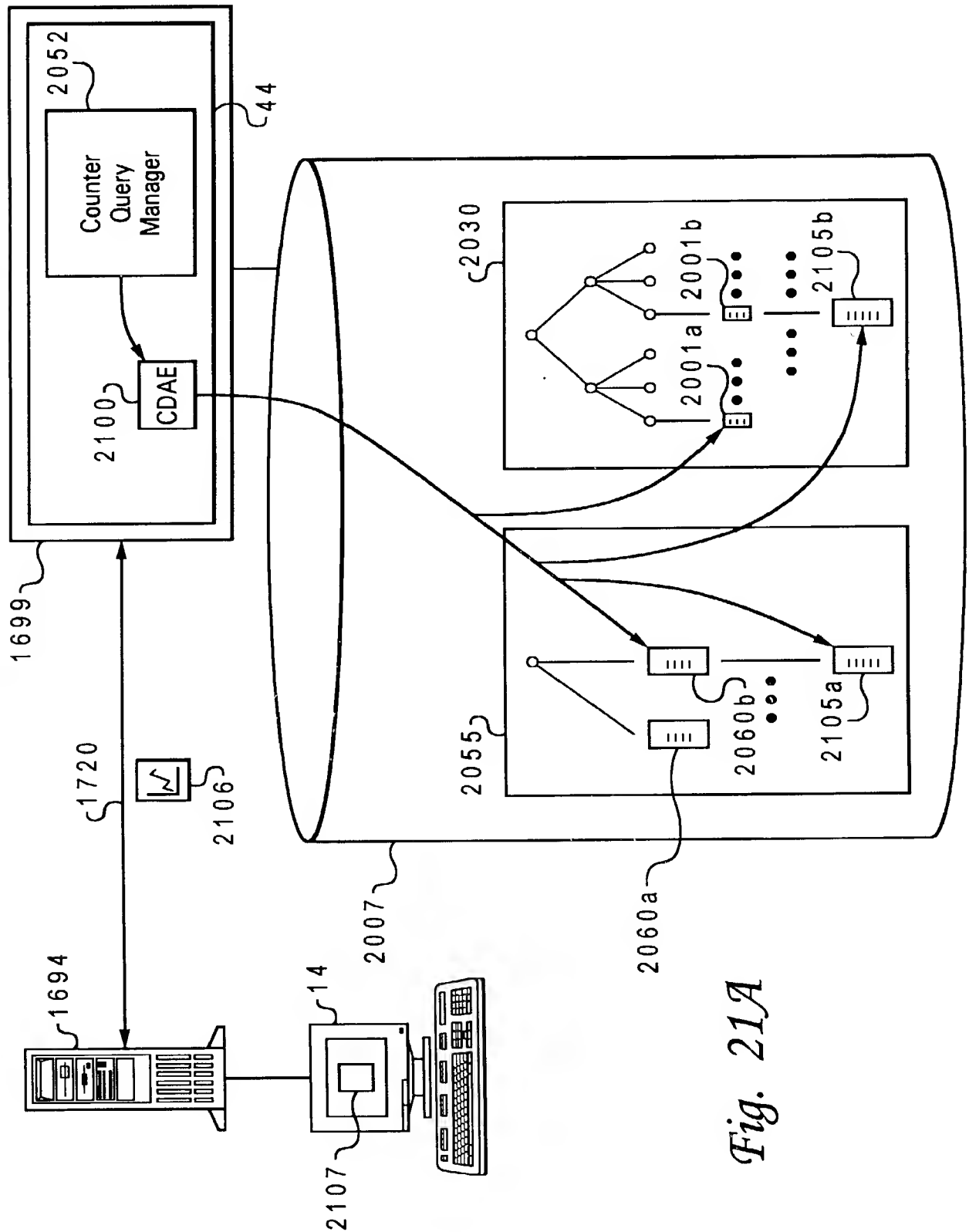


Fig. 21A

54/62

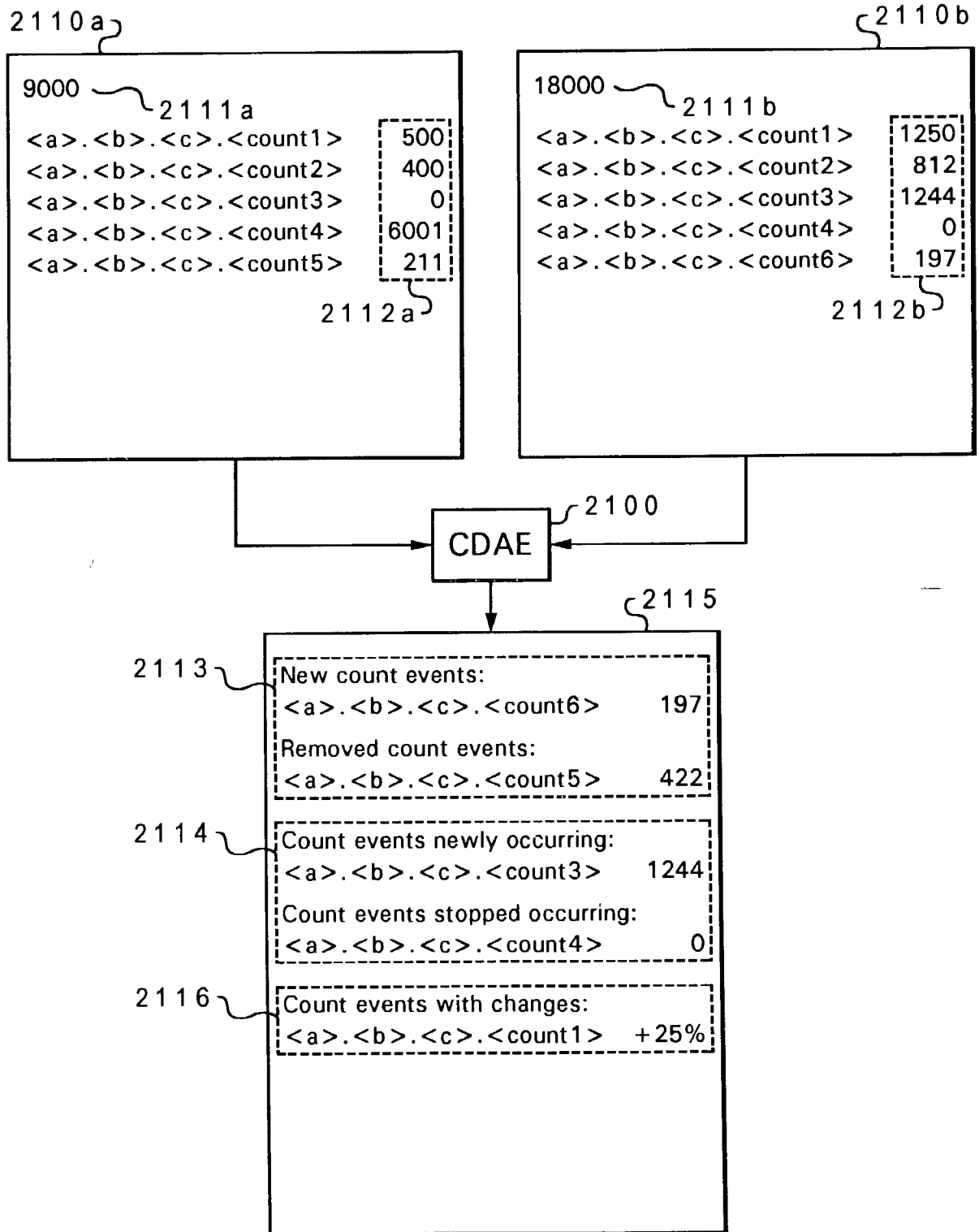


Fig. 21B

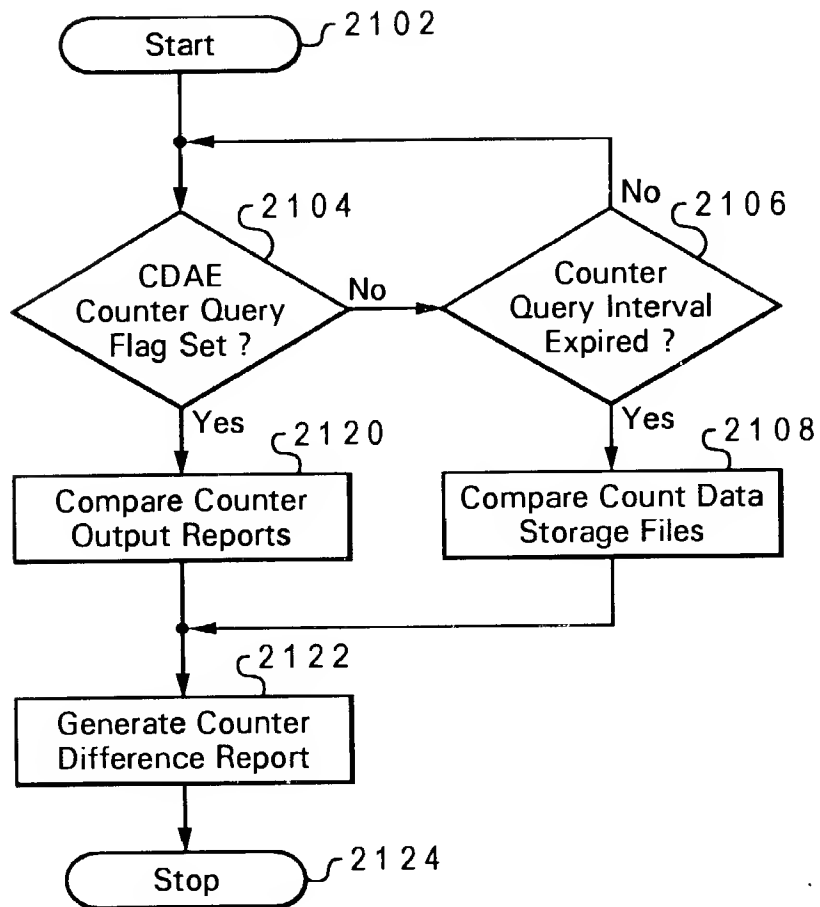
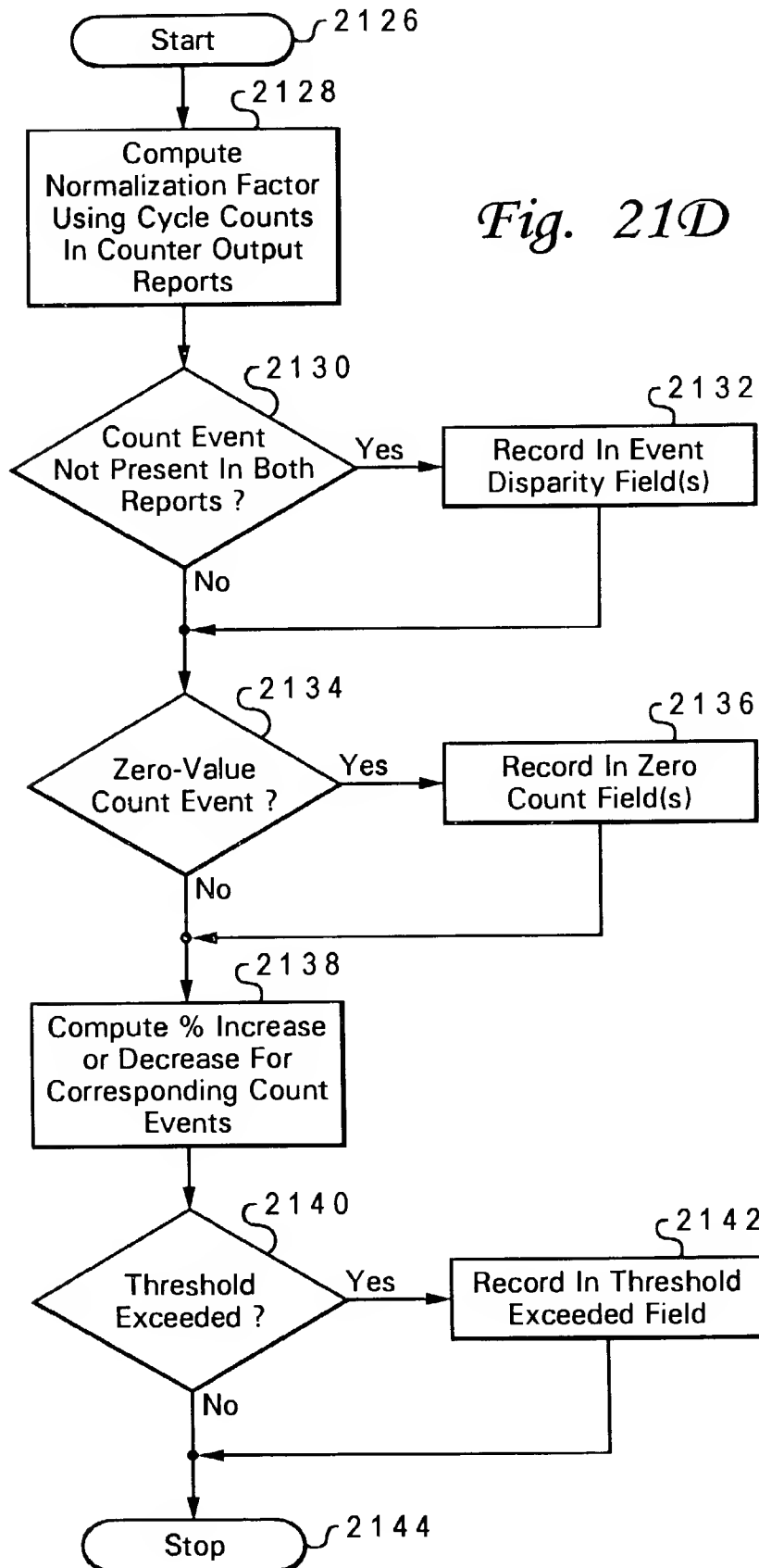
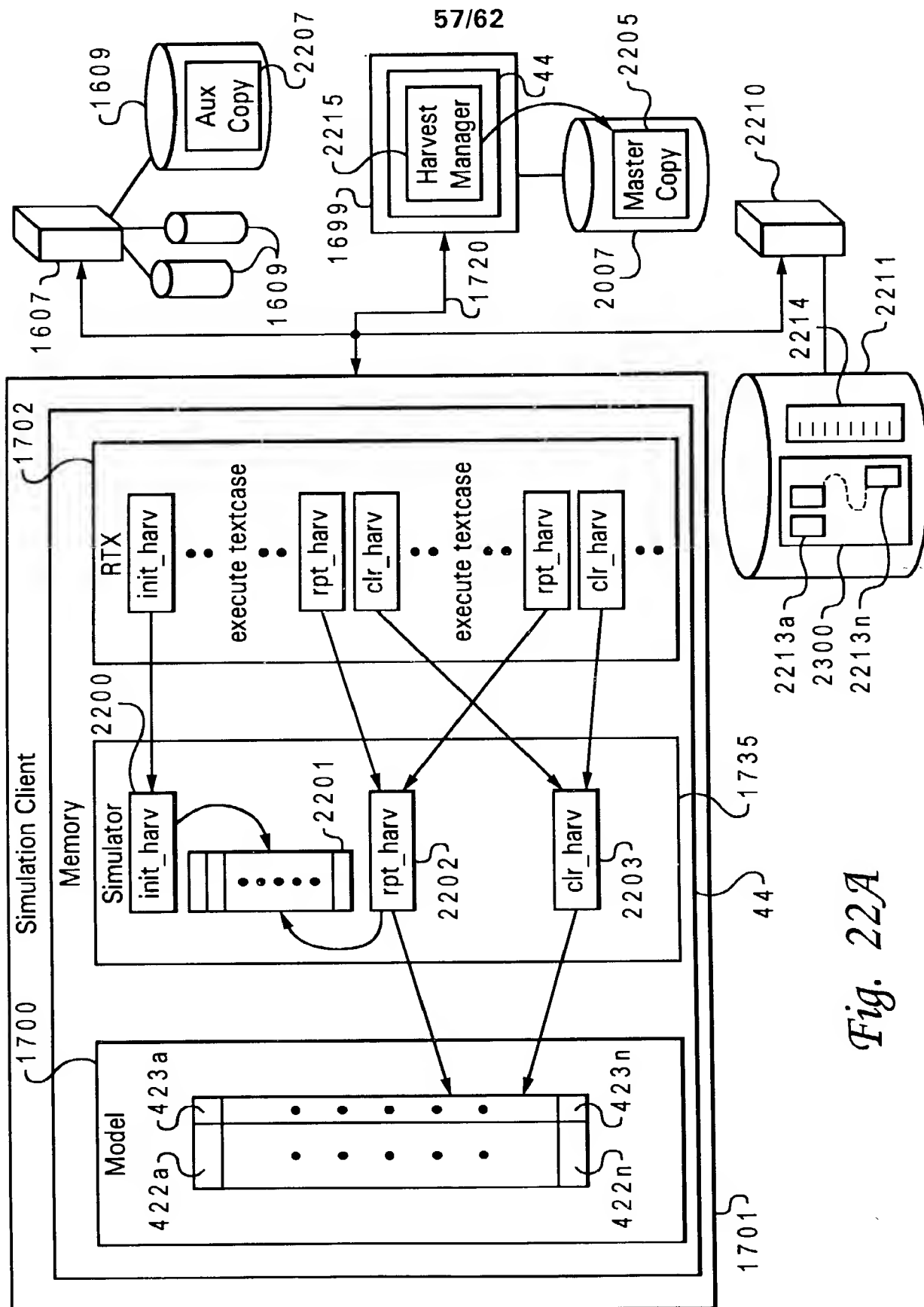


Fig. 21C





58/62

Fig. 22B

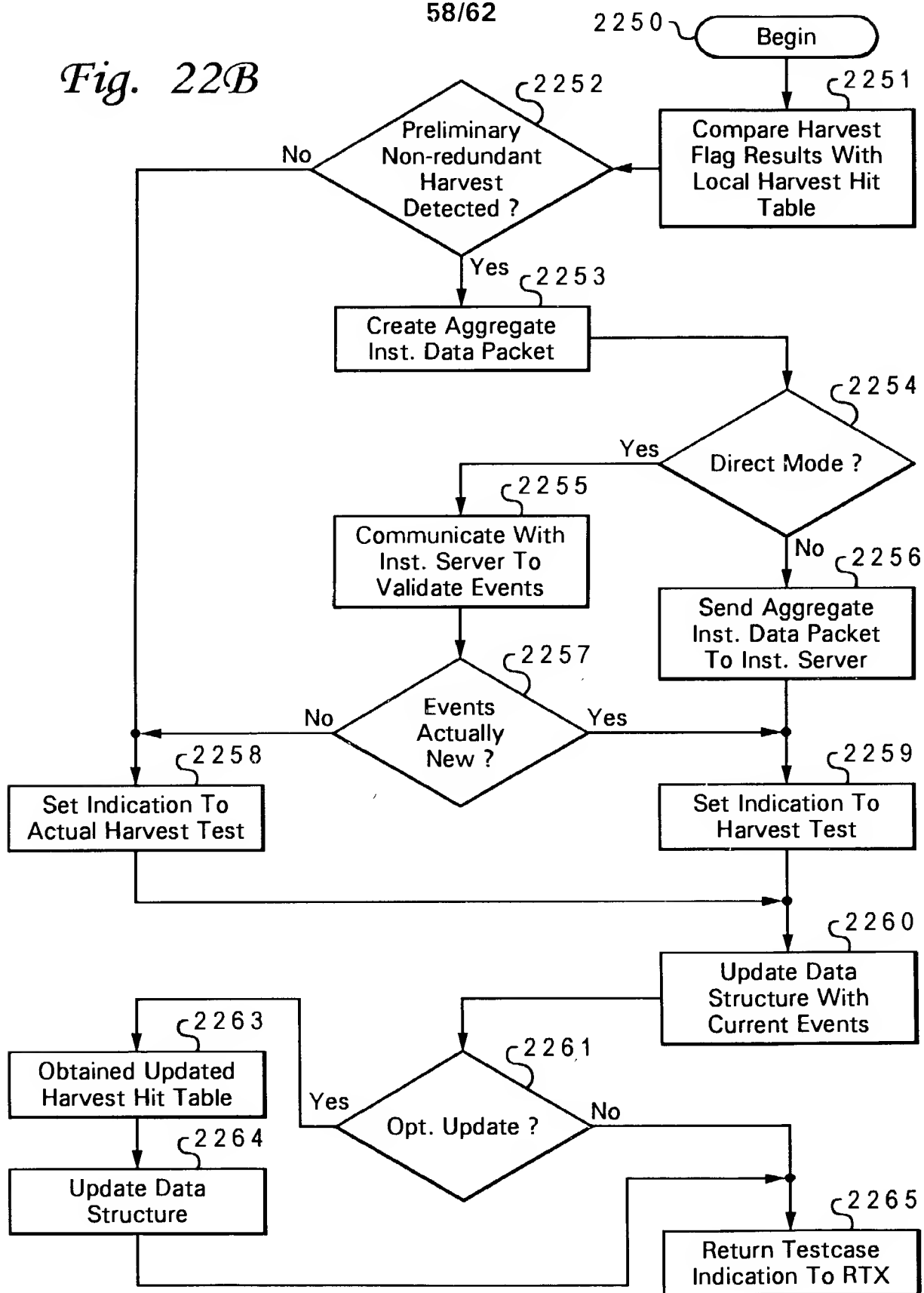
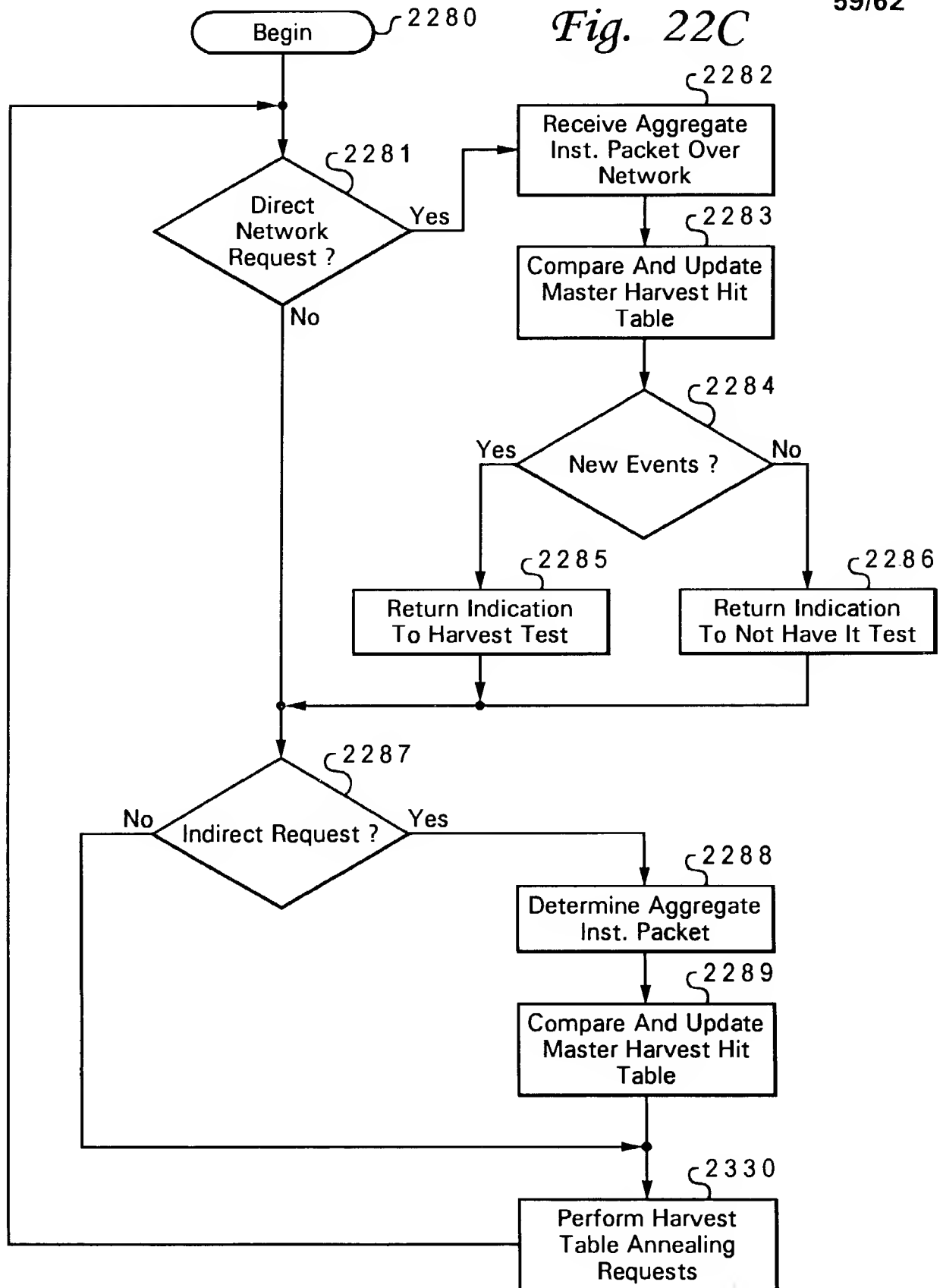


Fig. 22C



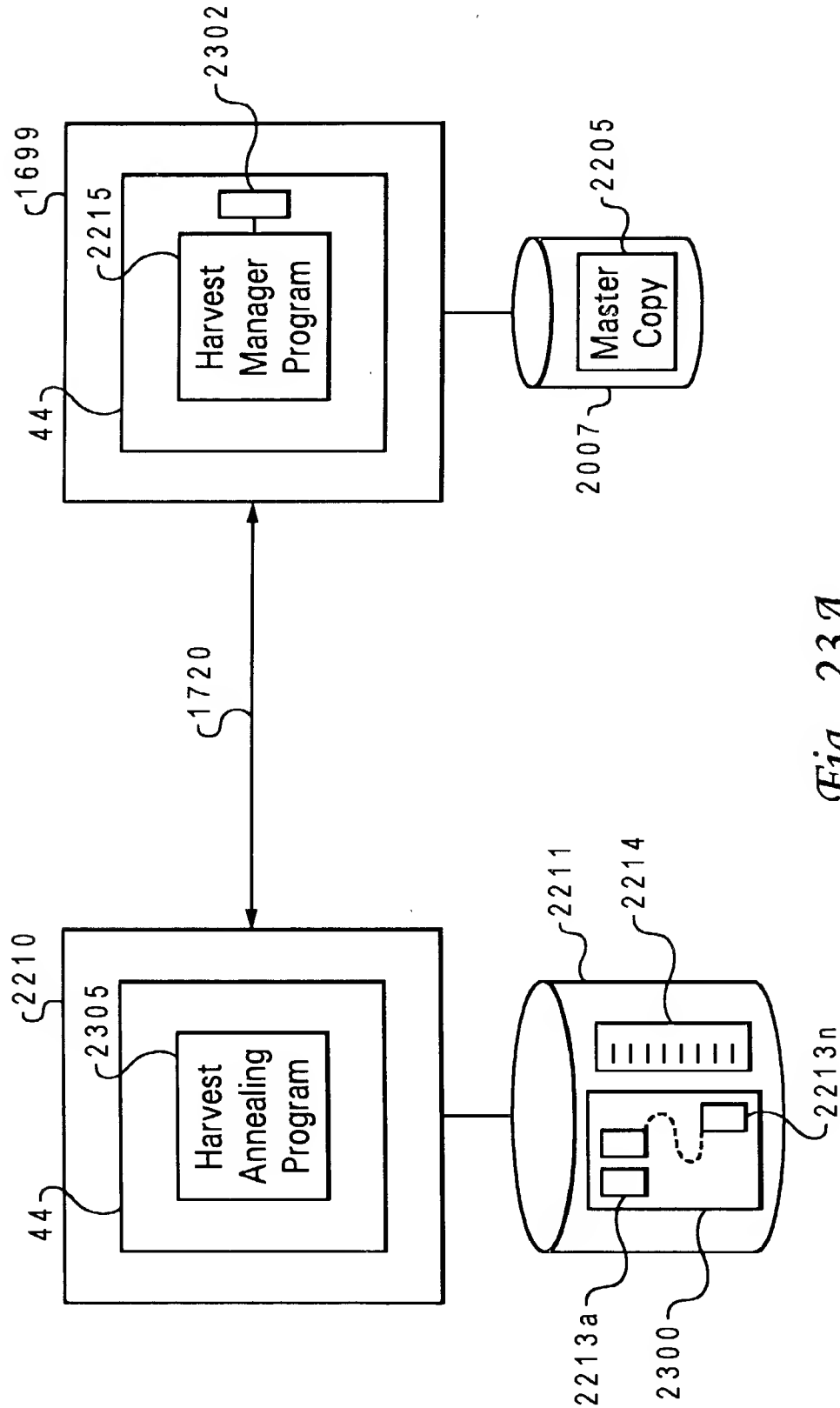


Fig. 23A

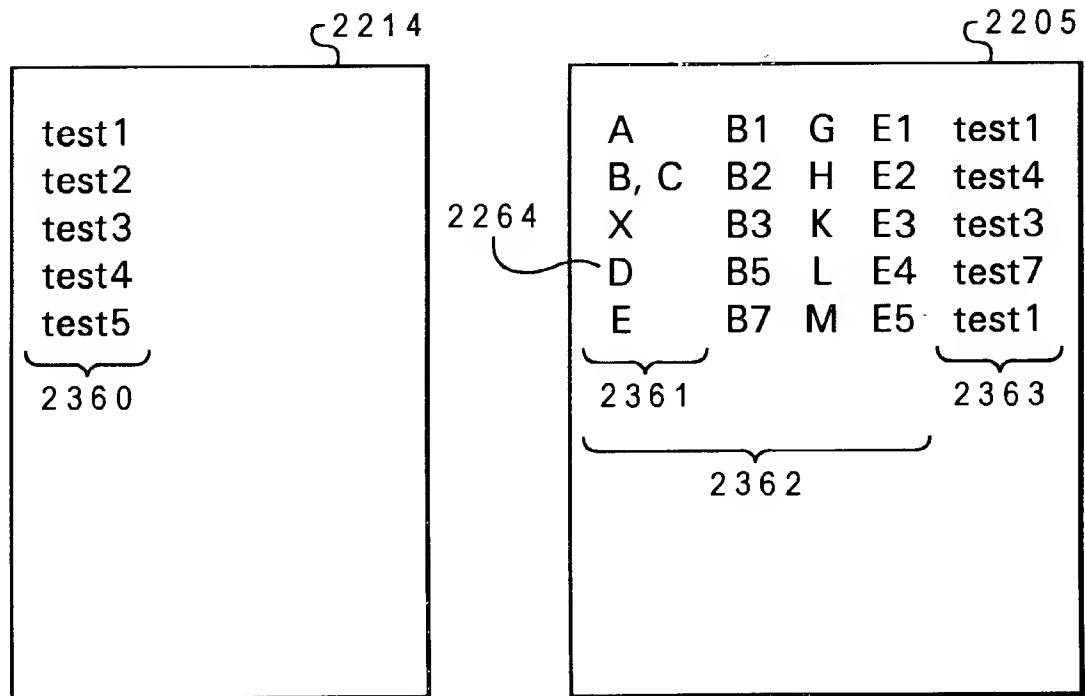


Fig. 23B

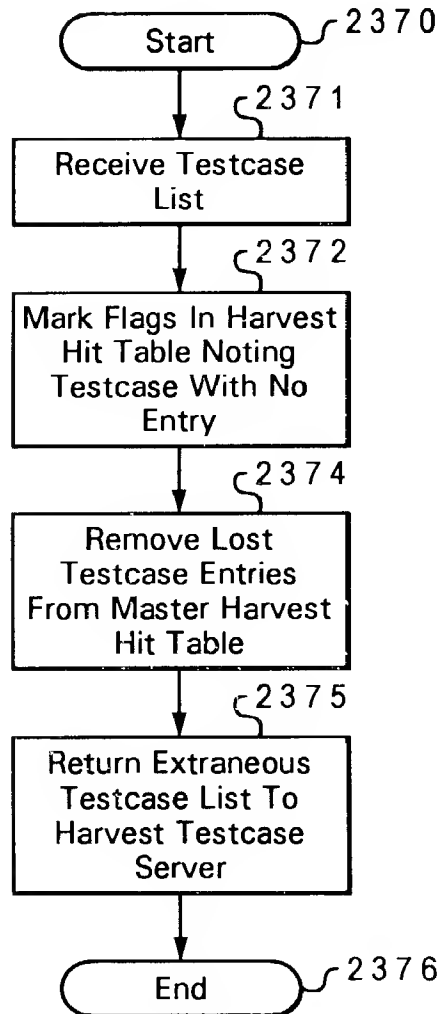


Fig. 23C